

Instituto Politécnico de Coimbra  
Instituto Superior de Engenharia de Coimbra  
Escola Superior de Tecnologia da Saúde Coimbra

---

# **Monitorização remota de utentes**

Filipe José Cera Fernandes

Mestrado em Sistemas e Tecnologias da Informação para a Saúde  
Coimbra, setembro, 2013



Instituto Politécnico de Coimbra  
Instituto Superior de Engenharia de Coimbra  
Escola Superior de Tecnologia da Saúde Coimbra

---

---

**Mestrado em Sistemas e Tecnologias da Informação para a Saúde**  
Projecto ou Estágio I e Projecto ou Estágio II

# **Monitorização remota de utentes**

Filipe José Cera Fernandes

**Orientador:**

**Eng.º Gustavo Corrente**

**Wise Systems**

**Co- Orientador:**

**Doutora Ana Rosa Borges**

**ISEC**

**Coimbra, setembro, 2013**

## Agradecimentos

Quero expressar aqui o meu agradecimento às pessoas que contribuíram para o sucesso deste projeto.

Agradeço especialmente aos orientadores de projeto, Doutora Ana Rosa Pereira Borges e ao Eng.º Gustavo Corrente por tão bem me terem orientado, assim como pelo interesse e incentivo que sempre demonstraram ao longo do decorrer de todo o projeto.

Finalmente desejo agradecer a todas as outras pessoas que de forma direta ou indiretamente contribuíram para a concretização deste projeto.

## Glossário

AJAX	Asynchronous Javascript and XML
AP	Access point
API	Application Programming Interface
CRUD	Create, Read, Update e Delete
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IDE	Integrated Development Environment
Json	JavaScript Object Notation
MAC	Media Access Control
MVC	Model-view-controller
ORM	Object Relational mapping
SGBD	Sistema de Gestão de Base de Dados
SMS	Short message service
SQL	Structured Query Language
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

## Índice

Agradecimentos.....	3
Glossário .....	4
Índice .....	5
Índice de figuras .....	8
Índice de tabelas.....	10
Resumo .....	11
Abstract .....	12
Capítulo 1. Introdução .....	13
1.1 Objetivos iniciais.....	14
1.2 Revisão da literatura e análise de outras soluções similares implementadas ....	16
Capítulo 2. Ferramentas e tecnologias .....	17
2.1 Ferramentas .....	17
2.2 Tecnologias.....	18
2.3 Template escolhido .....	20
Capítulo 3. Análise do sistema .....	21
3.1 Dispositivo .....	22
3.2 Esquema geral do sistema .....	23
3.3 Userstories .....	25
3.4 Análise da <i>Plataforma WISEcare</i> .....	27
3.4.1 Modelo de dados ( <i>Plataforma WISEcare</i> ) .....	27
3.4.2 Modelo físico de dados ( <i>Plataforma WISEcare</i> ).....	29
3.4.3 Descrição dos campos do modelo físico ( <i>Plataforma WISEcare</i> ).....	30
3.4.3.1 Tabela user .....	30
3.4.3.2 Tabela utent .....	30
3.4.3.3 Tabela utent_group.....	31
3.4.3.4 Tabela user_utent .....	31
3.4.3.5 Tabela configuration .....	31
3.4.3.6 Tabela configuration_params .....	31
3.4.3.7 Tabela event .....	32
3.4.3.8 Tabela event_notification .....	33
3.4.3.9 Tabela device.....	33
3.4.3.10 Tabela device_info .....	33

3.4.4	Mockups .....	35
3.5	Análise da Gateway WISEcare .....	42
3.5.1	Modelo de dados (Gateway WISEcare) .....	42
3.5.2	Modelo físico (Gateway WISEcare) .....	43
3.5.3	Descrição dos campos do modelo físico (Gateway WISEcare).....	44
3.5.3.1	Tabela Config.....	44
3.5.3.2	Tabela device.....	45
3.5.3.3	Tabela event .....	46
3.5.3.4	Tabela params .....	46
Capítulo 4.	Implementação do projeto.....	47
4.1	Recursos usados para implementação .....	47
4.1.1	Modelo de implementação – MVC.....	47
4.1.2	Modelo - Acesso a dados através do EBEAN .....	48
4.1.3	Definição das Vistas .....	49
4.1.4	Sistema de notificações .....	50
4.1.5	Interface.....	51
4.1.6	Implementação da <i>Plataforma WISEcare</i> .....	53
4.1.6.1	Autenticação .....	53
4.1.6.2	Dashboard .....	54
4.1.6.3	Dashboard – Evento .....	55
4.1.6.4	Eventos .....	56
4.1.6.5	Dispositivos .....	58
4.1.6.6	Grupos de utentes.....	60
4.1.6.7	Utentes .....	62
4.1.6.8	Utilizadores.....	65
4.1.6.9	Configurações.....	67
4.1.6.10	Alterar senha de utilizador .....	70
4.1.6.11	Suporte para multilíngue .....	71
4.1.6.12	Ajuda – Faqs .....	72
4.1.6.13	Sistema de notificações na plataforma.....	73
4.1.6.14	Colocação em produção.....	76
4.1.7	Implementação da Gateway WISEcare .....	77
4.1.7.1	Interface .....	78
4.1.7.2	Sincronização de configurações .....	85
4.1.7.3	Envio de eventos para a Plataforma WISEcare .....	87
4.1.7.4	Sistema de notificações.....	88

4.1.7.5	Instalação .....	89
Capítulo 5.	Conclusão e desenvolvimentos futuros .....	94
Bibliografia.....		96

## Índice de figuras

Figura 1 – Template utilizado .....	20
Figura 2 – Dispositivo utilizado no sistema .....	22
Figura 3 - Esquema geral do sistema .....	24
Figura 4 – Modelo de dados ( <i>Plataforma WISEcare</i> ) .....	27
Figura 5 - Modelo físico de dados ( <i>Plataforma WISEcare</i> ).....	29
Figura 6 – Login ( <i>mockup</i> ).....	35
Figura 7 – Dashboard ( <i>mockup</i> ).....	36
Figura 8 – Dispositivos ( <i>mockup</i> ) .....	37
Figura 9 – Utente ( <i>mockup</i> ) .....	38
Figura 10 – Utilizadores ( <i>mockup</i> ) .....	39
Figura 11 – Configurações ( <i>mockup</i> ) .....	40
Figura 12 - Parâmetros de configuração ( <i>mockup</i> ) .....	41
Figura 13 - Modelo de dados (Gateway WISEcare) .....	42
Figura 14 - Modelo físico(Gateway WISEcare) .....	43
Figura 15 - Modelo .....	48
Figura 16 - Vistas.....	49
Figura 17 - Exemplo de navegação .....	51
Figura 18 - Login .....	53
Figura 19 - Dashboard .....	54
Figura 20 Dashboard – Evento .....	55
Figura 21 - Eventos .....	56
Figura 22 - Eventos - visualizar .....	57
Figura 23 – Dispositivos .....	58
Figura 24 - Dispositivos - Editar .....	59
Figura 25 - Grupos de utentes .....	60
Figura 26 - Grupos de utentes – Novo/Editar .....	61
Figura 27 – Utentes .....	62
Figura 28 - Utente – Editar 1/3 .....	63
Figura 29 - Utente – Editar 2/3 .....	63
Figura 30 - Utente – Editar 3/3 .....	64
Figura 31 – Utilizadores .....	65
Figura 32 - Utilizadores - Editar .....	66
Figura 33 - Configuração.....	67
Figura 34 - Configuração – Edição 1/2.....	68
Figura 35 - Configuração - Edição 2/2 .....	69
Figura 36 - Mudar senha.....	70
Figura 37 – Suporte para multilíngue ( <i>Plataforma WISEcare</i> ) .....	71
Figura 38 - Ajuda – Faqs ( <i>Plataforma WISEcare</i> ).....	72
Figura 39- Nova mensagem (Json) .....	73
Figura 40 - Mensagem lida (Json).....	73
Figura 41 - Sistema de alertas .....	75
Figura 42 - Painel principal da Gateway WISEcare .....	78
Figura 43 – Configuração do Gateway WISEcare .....	79
Figura 44 - Editar configuração.....	81
Figura 45 - Parâmetros dos dispositivos.....	82
Figura 46 - Suporte para multilíngue ( <i>Gateway WISEcare</i> ).....	83



Figura 47 – Ajuda - Faqs ( <i>Gateway WISEcare</i> ) .....	84
Figura 48 - Mensagem Json pedido configurações .....	85
Figura 49 - Mensagem Json resposta configurações (sem configuração).....	85
Figura 50 - Mensagem Json envio configurações.....	86
Figura 51 - Mensagem Json envio evento para o servidor.....	87
Figura 52 - Mensagem resposta Json após inserir evento .....	87
Figura 53 - Mensagem Json com erro após inserir evento .....	87
Figura 54 - Mensagem Json com envio de um evento para um cliente web (browser) .....	88
Figura 55 - Colocar a directoria do Play na variável PATH do sistema .....	90
Figura 56 - Definir a variável JAVA_HOME .....	90
Figura 57 - Iniciar o serviço da <i>Plataforma WISEcare</i> .....	91
Figura 58 - Testar a plataforma .....	92
Figura 59 - Iniciar Gateway WISEcare.....	93
Figura 60 - Teste Gateway WISEcare .....	93

## Índice de tabelas

Tabela 1 - Tabela user ( <i>Plataforma WISEcare</i> ).....	30
Tabela 2 - Tabela utent ( <i>Plataforma WISEcare</i> ).....	31
Tabela 3 - Tabela utent_group ( <i>Plataforma WISEcare</i> ) .....	31
Tabela 4 - Tabela user_utent (relação m:n entre user e utent) ( <i>Plataforma WISEcare</i> ) .....	31
Tabela 5 - Tabela configuration ( <i>Plataforma WISEcare</i> ) .....	31
Tabela 6 - Tabela configuration_params ( <i>Plataforma WISEcare</i> ) .....	32
Tabela 7 - Tabela event ( <i>Plataforma WISEcare</i> ).....	32
Tabela 8 - Tabela event_notification ( <i>Plataforma WISEcare</i> ) .....	33
Tabela 9 - Tabela device ( <i>Plataforma WISEcare</i> ) .....	33
Tabela 10 - device_info ( <i>Plataforma WISEcare</i> ).....	34
Tabela 11 – Config ( <i>Gateway WISEcare</i> ) .....	44
Tabela 12 - Tabela device ( <i>Gateway WISEcare</i> ) .....	45
Tabela 13 - Tabela event ( <i>Gateway WISEcare</i> ) .....	46
Tabela 14 - Tabela configuration_params ( <i>Gateway WISEcare</i> ).....	46
Tabela 15 - Codificação dos níveis de severidade .....	74

## Resumo

Este trabalho tem como objetivo a análise e implementação de uma solução informática que permita monitorizar de forma permanente e fiável pessoas que necessitem de vigilância constante ou parcial.

Na fase inicial do projeto foi efetuado um levantamento de requisitos, identificando as necessidades encontradas para o sistema a desenvolver. Foi realizada uma análise de soluções semelhantes existentes no mercado e foram identificadas potenciais funcionalidades que ainda não existiam nas soluções encontradas.

Na fase seguinte foi identificado um conjunto de funcionalidades que tornariam o produto o mais completo possível. Das funcionalidades encontradas, foram escolhidas as que acrescentariam mais valor ao produto e que fossem realizáveis durante o tempo em que decorreu este projeto de mestrado. Essa escolha foi feita para que no final houvesse um produto funcional e com possibilidade de ser escalado para uma dimensão maior, adicionando mais funcionalidades.

Seguiu-se a escolha das ferramentas e tecnologias necessárias ao desenvolvimento do projeto, sendo que a mais relevante foi a plataforma play e o uso do motor de base de dados mySQL.

Na fase de análise foi decidida a arquitetura do projeto, foram identificados os dados que seriam necessários guardar no sistema e os atores que iriam interagir com o sistema. Foram igualmente identificadas as entidades e o seu relacionamento.

Na fase de implementação foram explorados vários modelos e técnicas para implementação das funcionalidades desejadas, assim como vários *templates* de forma a selecionar quais os que melhor se adaptariam ao produto que estava a ser desenvolvido.

## Abstract

This work aims at the analysis and implementation of a software solution that enables a permanent and reliable monitoring of people who need constant or partial supervision.

The project began with a survey of requirements, identifying which needs were necessary for the system to be developed. Then an analysis was conducted to find similar solutions in the market, identifying potential functionalities that didn't exist in the found solutions.

In the next phase it was identified a set of features that would allow the product to be as complete as possible. From this set of features, were chosen the ones that would give more value to the product and that could be achievable during the time reserved for this master's project. This choice was made in order to have a functional product with the possibility to scale to a higher dimension by adding more features.

The tools and technologies needed to develop the project were chosen before the implementation phase. The most important were the play framework and the use of the MySQL database engine.

During the analysis phase was decided the architecture design, which data would be stored in the system and the actors that who would interact with the system. Were also identified the entities and their relationships.

In the implementation phase various models and techniques, for implementing the desired functionality, have been explored, as well as several templates in order to select which ones would suit better to the project that was being developed.

## Capítulo 1. Introdução

A prestação de cuidados de saúde enfrenta atualmente um novo paradigma. As organizações de saúde, projetadas inicialmente para lidar com manifestações de doença aguda, têm agora de lidar cada vez mais com populações envelhecidas, com preponderância para contrair problemas relacionados com diabetes, insuficiência cardíaca, pressão arterial elevada (entre outros), necessitando de um novo tipo de cuidados prestados como: acompanhamento, monitorização e vigilância (Darkins & Sanders, 2009).

Uma vigilância contínua pode significar a diferença entre a vida e a morte de um paciente, em que as quedas são um dos maiores riscos para idosos que vivem sozinhos em casa, causando muitas vezes ferimentos severos. O risco é amplificado se a pessoa não puder pedir ajuda (Rougier *et al*, 2011).

Os sistemas de informação que permitem monitorizar e que ajudam a melhorar a qualidade de vida das pessoas estão a revolucionar a forma como está a ser fornecida assistência médica, apresentando muitas vantagens na sua utilização. As que mais se destacam são a possibilidade de o utente ser acompanhado na sua casa, permitindo redução de custos para o utente e em taxas hospitalares, uma intervenção mais rápida no caso de emergência, deteção precoce de situações de emergência e um elevado nível de segurança e confiança tanto para o utente como para as pessoas responsáveis por ele.

Neste sentido, com o sistema de informação desenvolvido neste projeto pretende-se ajudar os profissionais na área da saúde a monitorizar pessoas com mobilidade reduzida.

Através de um dispositivo portátil que é usado pela pessoa que está a ser monitorizada, é possível monitorizar a sua atividade e despoletar alertas caso a pessoa se encontre em dificuldades.

## 1.1 Objetivos iniciais

Este projeto tem como objetivo a análise de requisitos/funcionalidades e implementação de um sistema de informação de monitorização de pacientes/utentes. Com o projeto pretende-se obter uma solução que permita a monitorização de um conjunto de utentes que podem encontrar-se fisicamente distantes do seu cuidador e que possibilite uma intervenção rápida nas situações de emergência e/ou pedido de auxílio no dia-a-dia de uma instituição de saúde. Esta solução pode ser aplicada tanto em hospitais, clínicas, lares, como noutras instituições.<sup>1</sup>

Durante a fase de análise foram identificadas funcionalidades que tornariam o projeto mais completo, no entanto algumas dessas funcionalidades não foram implementadas, ficando como sugestões de trabalho futuro. As funcionalidades não implementadas não podem ser consideradas como alterações ao projeto, uma vez que ficou definido à partida que não seriam implementadas na fase inicial. Entre as várias funcionalidades a implementar em trabalho futuro estão as notificações por *sms*<sup>2</sup> ou *push*<sup>3</sup>, por exemplo.

O presente relatório está estruturado em cinco capítulos. Neste capítulo, “Introdução”, são descritas a motivação e os objetivos do trabalho realizado durante o projeto e é feito um pequeno resumo de cada um dos capítulos do relatório.

No capítulo 2, “Ferramentas e tecnologias”, são apresentadas as ferramentas e tecnologias escolhidas e usadas para a realização do projeto, assim como algumas considerações sobre o *template* escolhido para o interface da plataforma.

No capítulo 3 “Análise do sistema”, é exposto todo o trabalho inicial de análise, sendo dada uma explicação sobre a metodologia usada para a realização do projeto. Como foram aplicadas técnicas de *agile software development*, é apresentado um conjunto de *userstories* que serviram de base para identificar as funcionalidades a implementar.

---

<sup>1</sup> Nota: Toda a eletrónica envolvida no funcionamento do [dispositivo](#) não faz parte do âmbito deste projeto.

<sup>2</sup> SMS – Short message service (Wikipedia S. , 2013)

<sup>3</sup> Push message - Envio de mensagens através de uma conexão IP constantemente aberta para envio de notificação dos servidores para os dispositivos móveis

São também mostrados alguns esquemas de forma a melhorar a compreensão de todo o sistema. Em seguida é apresentado o modelo de dados com as várias entidades, e os seus relacionamentos.

Posteriormente é apresentado o modelo físico da base de dados com as várias tabelas e com a descrição de todos os campos e o tipo de dados usado.

Ainda na fase de análise, foram feitos vários *mockups*<sup>4</sup> com os vários ecrãs que a plataforma deveria ter, permitindo, assim, ficar com uma ideia do produto em funcionamento e das funcionalidades.

No capítulo 4, “Implementação do projeto” é apresentada a implementação. Expõe-se o modo como foi estruturado o trabalho e como as ferramentas apresentadas no capítulo anterior foram usadas para a implementação. Mostra-se também como é feito o acesso aos dados pela plataforma play, e é explicado como foram implementados os vários mecanismos de notificação.

Neste capítulo são, também, mostrados todos os ecrãs e todas as funcionalidades do projeto.

O capítulo 5 é reservado para as conclusões sobre o trabalho realizado, apresentando os objetivos iniciais e as tarefas realizadas para os atingir, assim como algumas pistas de desenvolvimento a realizar no futuro.

---

<sup>4</sup> *Mockup* é um protótipo não funcional que serve para demonstrar e testar o design. Podem servir para obter um feedback dos utilizadores ou para demonstrar uma funcionalidade. (Wikipedia M. , 2013)

## 1.2 Revisão da literatura e análise de outras soluções similares implementadas

Na pesquisa efetuada no início deste trabalho foram identificadas várias soluções que abordam o tema proposto. Destaque-se os sistemas de deteção de quedas por vídeo vigilância, sendo estes particularmente atrativos pelo fato dos utilizadores não necessitarem de usar equipamentos incomodativos. Apresentam, no entanto, algumas desvantagens, como a falta de privacidade e o desconforto por poderem ser vistos remotamente ou a ocorrência de falhas na área coberta pelo equipamento. Outro problema nos sistemas de deteção por vídeo é a necessidade de existir iluminação na área que está a ser monitorizada, sendo que uma variação de luminosidade pode provocar falsos positivos (Rougier et al., 2011)

(Gonçalves, 2008) refere no seu projeto de mestrado que foram identificados os requisitos para sensores médicos sem fios, dos quais se destacam:

- Portabilidade e não intrusão: dispositivos pequenos recolhem dados e comunicam sem fios, operam com o mínimo *input* possível do utilizador. A monitorização é feita no espaço em que vive;
- Operação permanente e em tempo real: os dados fisiológicos e ambientais recolhidos pelos sensores são feitos em tempo real e de forma contínua, permitindo uma resposta em tempo real;
- Privacidade dos registos e segurança: os dados pertencem aos pacientes no entanto os dispositivos sensores podem pertencer ao fornecedor de um serviço. O acesso aos dados deve ser controlado, guardando o registo de todos os acessos. Os dados devem estar sempre disponíveis.



## Capítulo 2. Ferramentas e tecnologias

### 2.1 Ferramentas

Para o desenvolvimento do projeto foram usadas as ferramentas que aqui se descrevem:

**Play! Framework** - Surgiu como uma *framework* Java, que fornecia suporte a linguagem Scala. Reconhecendo o poder e produtividade da linguagem Scala, a versão 2.0 do *framework* foi totalmente re-escrita em Scala. Atualmente, a Play Framework! 2.0 é um *framework* Scala, com suporte para Java. Uma das principais características é que se foca no desenvolvimento de um "*server-side stateless*". Isso facilita a escalabilidade das aplicações desenvolvidas na *framework* (Wikipedia - Scala, 2013).

**Eclipse** – É um IDE (*Integrated Development Environment*) de código aberto usado para desenvolvimento de *software*. Entre as suas muitas funcionalidades, destaca-se a facilidade de desenvolvimento em linguagem Java.

**Firefox** - É um navegador livre, multi-plataforma desenvolvido pela *Mozilla Foundation*. Atualmente vai na versão 12 e é o 3º navegador mais usado em todo o mundo.

**Firebug** – É uma *plug in* para o *firefox* que permite ver e alterar em tempo real documentos html, css e javascript. Permite efetuar *debug* e monitorizar o tempo de carregamento de uma página.

**Chrome** – É o navegador mais usado em todo o mundo. Foi desenvolvido pela *Google*. Possui ferramentas de análise de *html*, *css* e *javascript*.

**mySQL** – É um Sistema de Gestão de Base de Dados (SGBD) que usa linguagem SQL como interface. É um motor muito popular e é amplamente usado por todo o mundo. Este é dos motores mais suportados pelos serviços de alojamento (MYSQL, 2013).

**Subversion** - *Apache Subversion* (também conhecido por *svn*) é um sistema de controlo de versão com histórico de revisões.

## 2.2 Tecnologias

Para a realização deste projeto foi escolhida a Plataforma *Play* (*Play Framework!*). O *play* é uma plataforma altamente produtiva de desenvolvimento em Java e Scala que integra componentes e API's necessários para desenvolvimento de aplicações modernas. O *Play* é baseado numa arquitetura leve, sem estado (*stateless*), com consumo de recursos mínimo e previsível (CPU, memória, *threads*) para aplicações altamente escaláveis (*Play framework!*, 2013).

**Java** – Java é uma linguagem de programação e uma plataforma de computação lançada pela primeira vez pela *Sun Microsystems* em 1995. É a tecnologia que capacita muitos programas da mais alta qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros, por exemplo. O Java é executado em mais de 850 milhões de computadores pessoais e em bilhões de dispositivos em todo o mundo, inclusive telefones celulares e dispositivos de televisão (Oracle, 2013).

**HTML** - Acrónimo para a expressão inglesa *HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto. HTML é o conjunto de especificações (símbolos) que determinam como o *browser* formatará o texto e qual a função que cada parte do texto terá no site.

**CSS** – *Cascading Style Sheets* (ou simplesmente CSS) é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos numa linguagem de marcação, como HTML ou XML. O seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

**SQL** - *Structured Query Language*, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Muitas das características originais do *SQL* foram inspiradas na álgebra relacional.

**Javascript** - é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que *scripts* pudessem ser executados do lado do cliente e interagissem com o utilizador sem a necessidade deste *script* passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

É atualmente a principal linguagem para programação *client-side* em navegadores web (Javascript, 2013).

## 2.3 Template escolhido

Para o desenvolvimento desta plataforma optou-se por usar um *template* pré-construído. Esse pode ser obtido por 16 USD em <http://jumpstartui.com/> e fornece um aspeto profissional à plataforma, que de outra forma seria muito mais trabalhoso de conseguir.

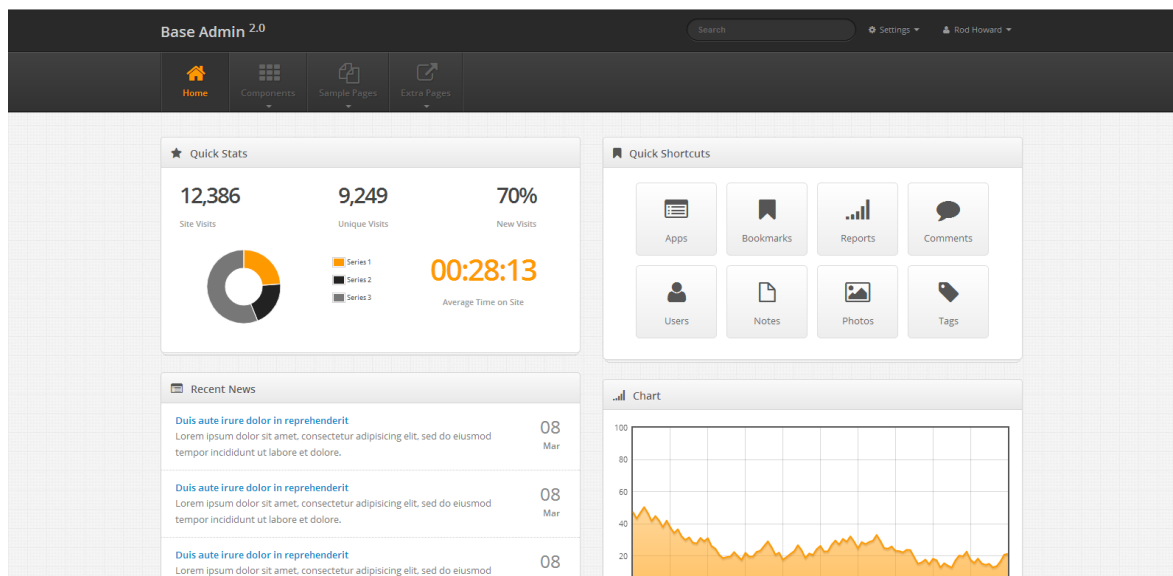


Figura 1 – Template utilizado

## Capítulo 3. Análise do sistema

Para o desenvolvimento deste projeto foram aplicadas técnicas *agile software development*. Em primeiro lugar foram identificadas as funcionalidades que são necessárias num sistema de monitorização remota de utentes. Em seguida, o sistema foi descrito em formato de *userstories*, como mostrado na seção 3.3. Posteriormente, e com base nas *userstories*, foi desenhado o modelo de dados que é apresentado nas secções 3.4.1 a 3.4.3 e nas secções 3.5.1 a 3.5.3.

Após o desenho do modelo de dados foi feita a modelação do sistema em formato de *mokups* que é apresentado na secção 3.4.4.

## 3.1 Dispositivo

O dispositivo (Figura 2) apresenta as seguintes características:

- Tem a capacidade de se ligar através da rede *wireless* a um serviço que se encontra a correr na rede local;
- Possui um botão de pânico que permite a um utente chamar ajuda;
- Envia periodicamente informação “alive” com a localização atual e o estado do dispositivo;
- Envia alerta no caso de ser removido do “clip” – o dispositivo possui um sensor para verificar se está colocado no cinto do utente ou se foi removido;
- Possui mecanismo de deteção de quedas através de acelerómetro, enviando uma mensagem caso detete uma queda;



Figura 2 – Dispositivo utilizado no sistema

## 3.2 Esquema geral do sistema

O sistema apresentado é composto por dois componentes que em conjunto permitem obter um maior leque de funcionalidades.

O componente designado como *Plataforma WISEcare* é uma aplicação web que concentra toda a lógica e regras de negócio relacionada com a gestão de utentes.

O componente designado como *Gateway WISEcare* é também uma aplicação web que concentra a lógica de controlo do sistema de comunicação com os dispositivos.

As duas aplicações operam em conjunto e comunicam entre si trocando mensagens de configuração e alerta.

O sistema foi desenhado de forma a permitir uma grande flexibilidade na sua utilização. Assim, prevêem-se 3 tipos de cenários:

1. Pode ser aplicado de forma organizada em instalações de grandes dimensões, encontrando-se preparado para gerir um grande número de utentes, como é o caso de um hospital, um centro de saúde ou um lar de idosos. É usado um servidor com a *Plataforma WISEcare* e um servidor com o *Gateway WISEcare* (ambos os serviços podem partilhar a mesma máquina);
2. Pode ser aplicado num serviço de *call center* em que existe uma monitorização concentrada e remota de utentes que se encontram em suas casas. Com este *setup* existe um serviço com a *Plataforma WISEcare* e vários serviços de *Gateway WISEcare*, cada um instalado na habitação do utente;
3. Utilização apenas como *gateway* sem necessidade de gestão de utentes. Para utilização em locais com poucos utentes em que não se torna necessário registar informação dos mesmos. Este cenário não requer instalação de servidor com a *Plataforma WISEcare*, bastando apenas configurar a *gateway* como *stand alone*.

O esquema na Figura 3 apresenta uma visão geral do sistema.

Este projeto é composto por um ou vários dispositivos e um *gateway* que recebe ligações UDP, ligados na mesma rede e cada *gateway* corresponde a um espaço físico distinto que apenas tem comunicação com a *Plataforma WISEcare*. Dentro de cada espaço físico pode existir um ou vários dispositivos que comunicam com o *gateway*. Neste servidor local, chamado de *Gateway WISEcare*, existe um serviço que fica à “escuta” numa determinada porta à espera de receber ligações UDP (tipicamente será na porta 5000, sendo no entanto configurável). Esse serviço recebe e pré-processa as mensagens dos dispositivos que, dependendo do tipo de mensagem, comunicam imediatamente com o serviço web, chamado de *Plataforma WISEcare* a fim de enviar um aviso aos vários clientes web (monitores).

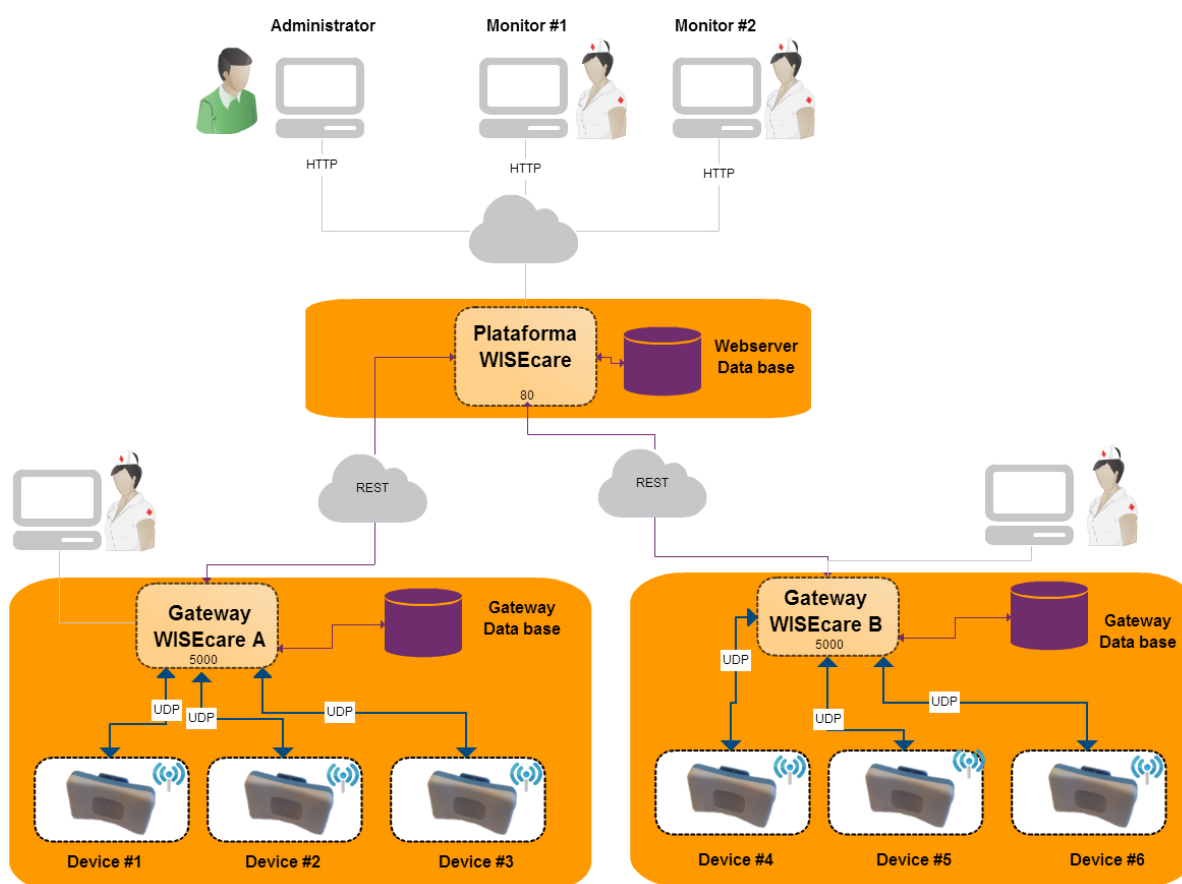


Figura 3 - Esquema geral do sistema



## 3.3 Userstories

Com o objetivo de identificar as funcionalidades que o sistema deve possuir, estas foram descritas em forma de “userstories”. Existem várias vantagens em usar o formato de userstories para descrever as funcionalidades do sistema; nomeadamente: com este formato é possível saber “quem”, o “quê” e o “porquê” de cada funcionalidade e assim, associar cada “userstorie” a uma tarefa a ser desenvolvida. As “userstories” estão divididas em duas seções: dispositivo e aplicação.

### Dispositivo

- #1 Como administrador quero poder criar, alterar e consultar dispositivos;
- #2 Como administrador quero criar, editar e consultar uma configuração de dispositivo;
- #3 Como monitor quero poder visualizar o estado atual de todos os dispositivos;
- #4 Como sistema quero que seja gerado um evento quando é recebida uma mensagem de um dispositivo;

### Aplicação

- #5 Como monitor quero associar dispositivos a utentes;
- #6 Como monitor quero receber um aviso quando um utente pede auxílio;
- #7 Como monitor quero poder visualizar o estado de todos os utentes/dispositivos;
- #8 Como administrador quero poder criar, alterar e consultar utentes;
- #9 Como administrador de sistema quero poder criar, alterar e consultar outros utilizadores;
- #10 Como monitor quero configurar o dispositivo de um utente, aplicando uma configuração ou alterando um parâmetro;
- #11 Como monitor quero consultar o histórico de um dispositivo/utente;
- #12 Como monitor quero receber um alerta sempre que há um evento crítico;
- #13 Como monitor quero dispensar um alerta;
- #14 Como monitor quero poder configurar um horário para cada perfil de configuração;
- #15 Como monitor quero configurar uma zona em que o dispositivo tenha um comportamento diferente;
- #16 Como monitor quero receber notificações via SMS, Email, Push;
- #17 Como monitor quero receber notificações apenas dos meus utentes;

#18 Como administrador quero que o sistema alerte outras pessoas quando um alerta não for dispensado no período de tempo definido.

Nesta fase do projeto a identificação e descrição das *userstories* foi importante, uma vez que ajudou a definir as funcionalidades e restrições do projeto.

### 3.4 Análise da Plataforma WISEcare

#### 3.4.1 Modelo de dados (Plataforma WISEcare)

Nesta fase, e com base nas funcionalidades a serem desenvolvidas, foram identificadas as entidades intervenientes no sistema em termos de dados e a respetiva relação entre elas. Neste diagrama, Figura 4, não são apresentados todos os campos existentes, pretendendo-se, apenas, apresentar uma vista global sobre as várias entidades e como se relacionam entre si.

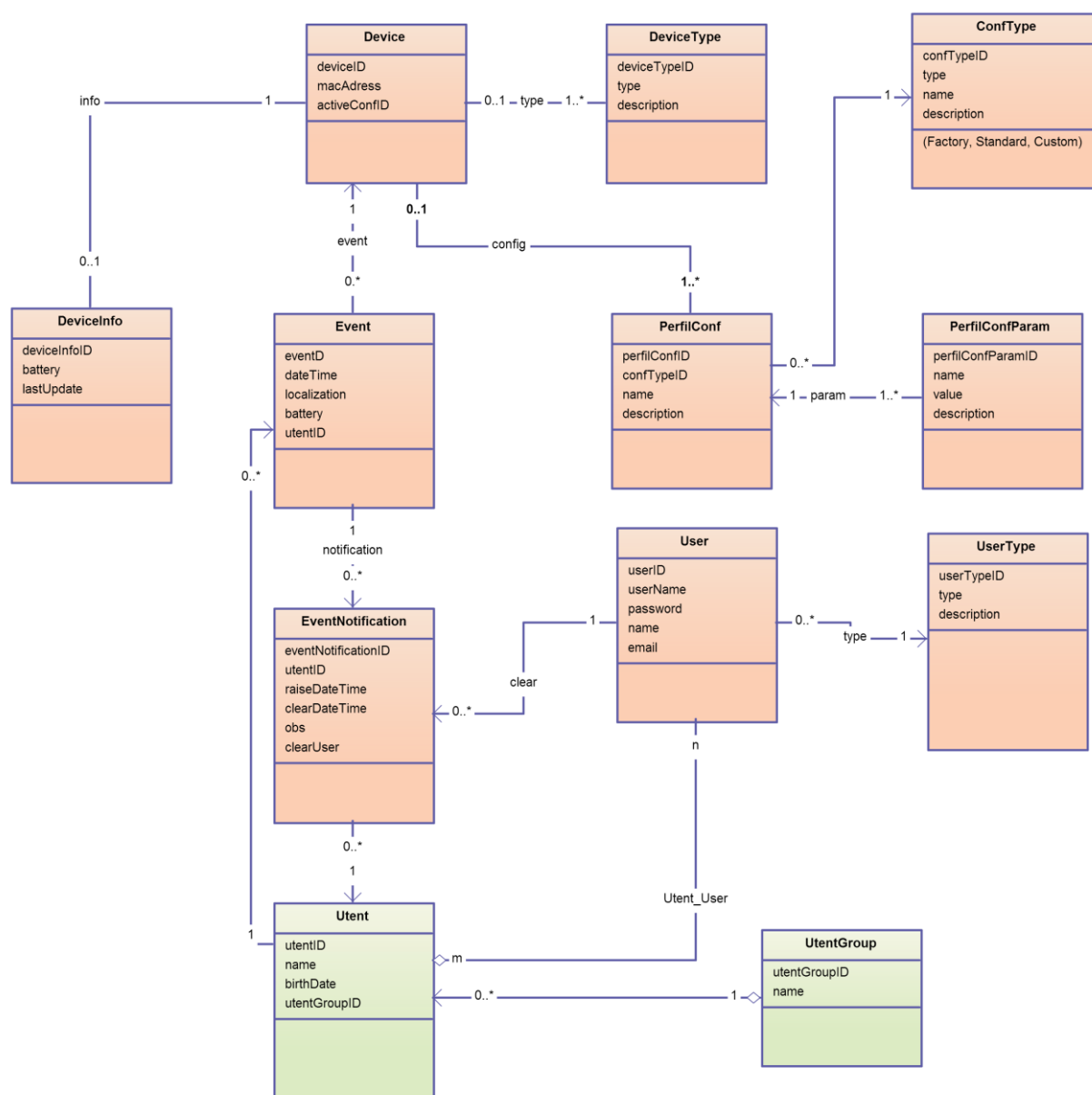


Figura 4 – Modelo de dados (Plataforma WISEcare)

As entidades Utent e UtentGroup, embora façam parte do sistema e estejam relacionadas com outras entidades do sistema, estão marcadas com cor diferente por representarem dados relativos a Utentes (pessoas monitorizadas pelo sistema).

### 3.4.2 Modelo físico de dados (Plataforma WISEcare)

A Figura 5 é uma representação do modelo físico da base de dados e serviu de base para as tabelas que são apresentadas nos pontos seguintes.

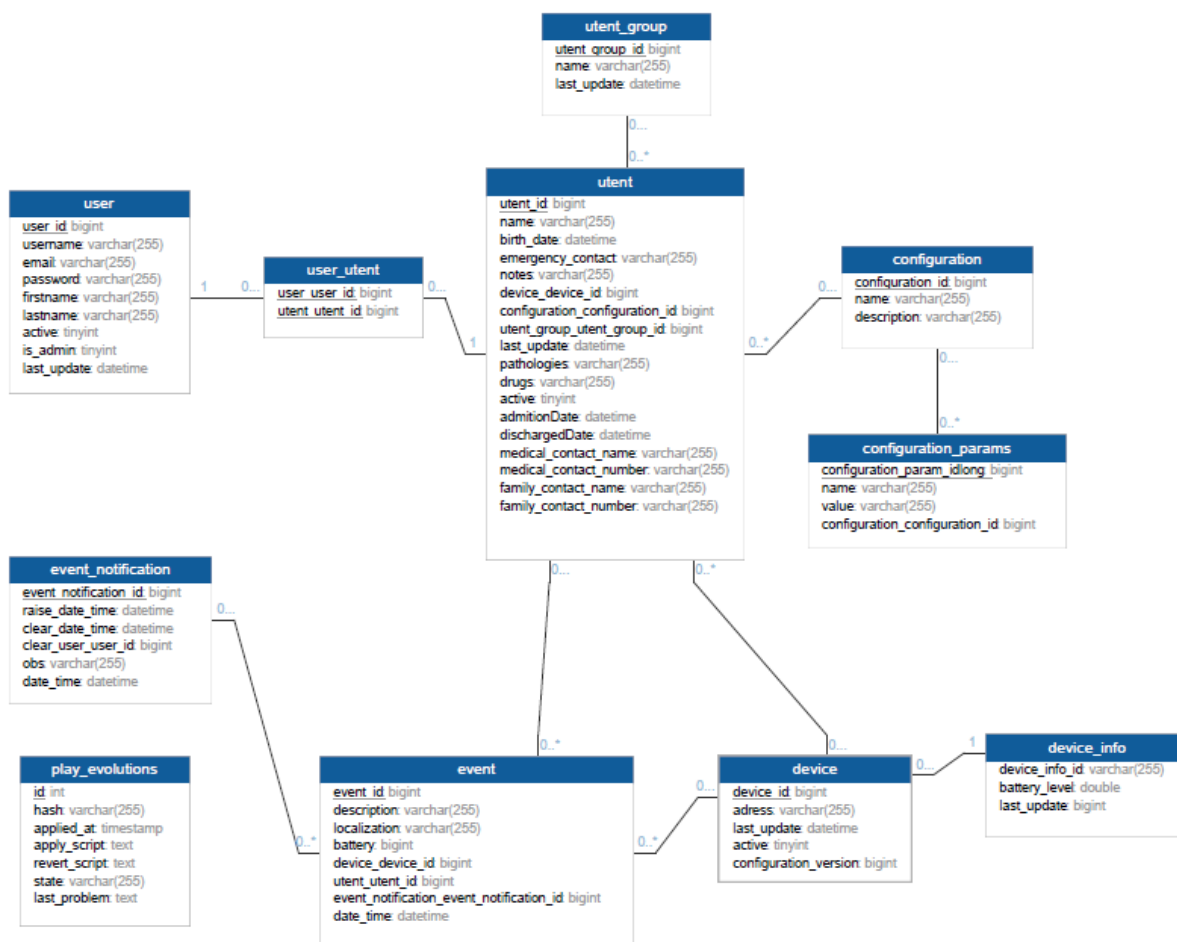


Figura 5 - Modelo físico de dados (Plataforma WISEcare)

### 3.4.3 Descrição dos campos do modelo físico (*Plataforma WISEcare*)

Nesta seção do relatório irá proceder-se à descrição das tabelas do modelo físico, campo a campo.

#### 3.4.3.1 Tabela user

A tabela *user* (Tabela 1) armazena os dados dos utilizadores do sistema.

Tabela user		
Nome campo	Tipo campo	Descrição
user_id	bigint	Chave primária
username	varchar(255)	Nome do utilizador
email	varchar(255)	Email do utilizador
password	varchar(255)	Senha do utilizador
firstname	varchar(255)	Primeiro nome do utilizador
lastname	varchar(255)	Último nome do utilizador
active	tinyint(1)	Indica se o utilizador se encontra ativo no sistema
last_update	datetime	Última atualização do registo
user_type	int	Tipo de utilizador (enumerado no Java Supervisor, Monitor, Audit)

Tabela 1 - Tabela user (*Plataforma WISEcare*)

#### 3.4.3.2 Tabela utent

A tabela *utent* (Tabela 2) armazena os dados dos utentes do sistema. Relaciona-se com as tabelas *device* para saber qual é o dispositivo atualmente associado ao utente, *utent\_group* (com o grupo do utente) e *configuration* para guardar a configuração atual do dispositivo associado ao utente.

Tabela utent		
Nome campo	Tipo campo	Descrição
utent_id	bigint	Chave primária
name	varchar(255)	Nome do utente
birth_date	datetime	Data de nascimento do utente
emergency_contact	varchar(255)	Contacto de emergência
emergency_contact_name	varchar(255)	Nome do contacto de emergência
notes	varchar(255)	Notas do utente
device_device_id	bigint	Chave estrangeira da tabela <i>device</i>
configuration_configuration_id	bigint	Chave estrangeira da tabela <i>configuration</i>
utent_group_utent_group_id	bigint	Chave estrangeira da tabela <i>utent_group</i>
last_update	datetime	Última atualização do registo
pathologies	varchar(255)	Doenças do utente
drugs	varchar(255)	Medicamentos usados pelo utente
active	tinyint(1)	Indica se o utente se encontra

		ativo no sistema
admissionDate	datetime	Data de admissão
dischargedDate	datetime	Data de alta
medical_contact_name	varchar(255)	Nome do contato médico
medical_contact_number	varchar(255)	Número do contato médico
family_contact_name	varchar(255)	Nome de familiar a contatar
family_contact_number	varchar(255)	Numero do familiar a contatar
identification_number	varchar(255)	Número único de identificação

Tabela 2 - Tabela utent (Plataforma WISEcare)

#### 3.4.3.3 Tabela utent\_group

A tabela *utent\_group* (Tabela 3) guarda a informação dos grupos de utentes.

Tabela utent_group		
Nome campo	Tipo campo	Descrição
utent_group_id	bigint	Chave primária
Name	varchar(255)	Nome do grupo
last_update	datetime	Última atualização do registo

Tabela 3 - Tabela utent\_group (Plataforma WISEcare)

#### 3.4.3.4 Tabela user\_utent

A tabela *user\_utent* (Tabela 4) surge da relação de m:n das entidades *user* e *utent*.

Tabela user_utent		
Nome campo	Tipo campo	Descrição
user_user_id	bigint	Chave estrangeira da tabela <i>user</i>
utent_utent_id	bigint	Chave estrangeira da tabela <i>utent</i>

Tabela 4 - Tabela user\_utent (relação m:n entre user e utent) (Plataforma WISEcare)

#### 3.4.3.5 Tabela configuration

A tabela *configuration* (Tabela 5) serve para guardar uma configuração genérica de um dispositivo. Associada à configuração, existe a tabela *configuration\_params* com os parâmetros da configuração.

Tabela configuration		
Nome campo	Tipo campo	Descrição
configuration_id	bigint	Chave primária
name	varchar(255)	Nome da configuração
description	varchar(255)	Descrição da configuração

Tabela 5 - Tabela configuration (Plataforma WISEcare)

#### 3.4.3.6 Tabela configuration\_params

A tabela *configuration\_params* (Tabela 6) guarda os valores dos parâmetros de uma configuração.

Tabela configuration_params		
Nome campo	Tipo campo	Descrição
configuration_param_id	bigint	Chave primária
name	varchar(255)	Nome do parâmetro
value	varchar(255)	Valor do parâmetro
configuration_configuration_id	bigint	Chave estrangeira tabela <i>configuration</i>

Tabela 6 - Tabela configuration\_params (Plataforma WISEcare)

### 3.4.3.7 Tabela event

A tabela *event* (Tabela 7) guarda os vários eventos que provêm dos dispositivos ligados ao sistema. Contém informação da data, hora, localização do dispositivo, informação sobre o estado da bateria, a descrição do evento e utente associado. Relaciona com a tabela *event\_notification* para determinar se o evento foi visualizado e/ou dispensado por um utilizador do sistema.

Tabela event		
Nome campo	Tipo campo	Descrição
event_id	bigint	Chave primária
description	varchar(255)	Nome da configuração
localization	varchar(255)	Descrição da configuração
battery	bigint	Estado da bateria do dispositivo
device_device_id	bigint	Chave estrangeira tabela <i>device</i>
utent_utent_id	bigint	Chave estrangeira tabela <i>utent</i>
event_notification_event_notification_id	bigint	Chave estrangeira tabela <i>event_notification</i>
date_time	datetime	Última atualização do registo
event_type	integer	Tipo de evento (Enumerado no Java – UtentMoving, OutOfRange, PanicButton, etc)
gravity	integer	Nível de gravidade – É usado para mostrar cores diferentes no dashboard e permite facilmente identificar uma situação perigosa

Tabela 7 - Tabela event (Plataforma WISEcare)



### 3.4.3.8 Tabela event\_notification

A tabela *event\_notification* (Tabela 8) armazena a notificação de um evento, quando foi gerado, quando foi lido e qual foi o utilizador.

Tabela event_notification		
Nome campo	Tipo campo	Descrição
event_notification_id	bigint	Chave primária
raise_date_time	datetime	Data/Hora do momento em que ocorreu o evento
clear_date_time	datetime	Data/Hora do momento em que o evento foi lido/limpo
clear_user_user_id	bigint	Chave estrangeira da tabela <i>user</i> com o utilizador que leu ou limpou o evento
obs	varchar(255)	Observação inserida pelo utilizador quando leu a notificação
date_time	datetime	Última atualização do registo

Tabela 8 - Tabela event\_notification (Plataforma WISEcare)

### 3.4.3.9 Tabela device

A tabela *device* (Tabela 9) armazena a informação dos dispositivos registados no sistema.

Tabela device		
Nome campo	Tipo campo	Descrição
device_id	bigint	Chave primária
adress	varchar(255)	Endereço MAC <sup>5</sup>
last_update	datetime	Descrição da configuração
active	tinyint(1)	Indica se o dispositivo se encontra ativo no sistema
configuration_version	bigint	Versão da configuração – Serve para a sincronização de parâmetros

Tabela 9 - Tabela device (Plataforma WISEcare)

### 3.4.3.10 Tabela device\_info

A tabela *device\_info* (Tabela 10) armazena a informação sobre a última informação do dispositivo.

Tabela device_info		
Nome campo	Tipo campo	Descrição
device_info_id	bigint	Chave primária
battery_level	double	Nível de bateria do dispositivo
last_update	datetime	Última atualização do registo

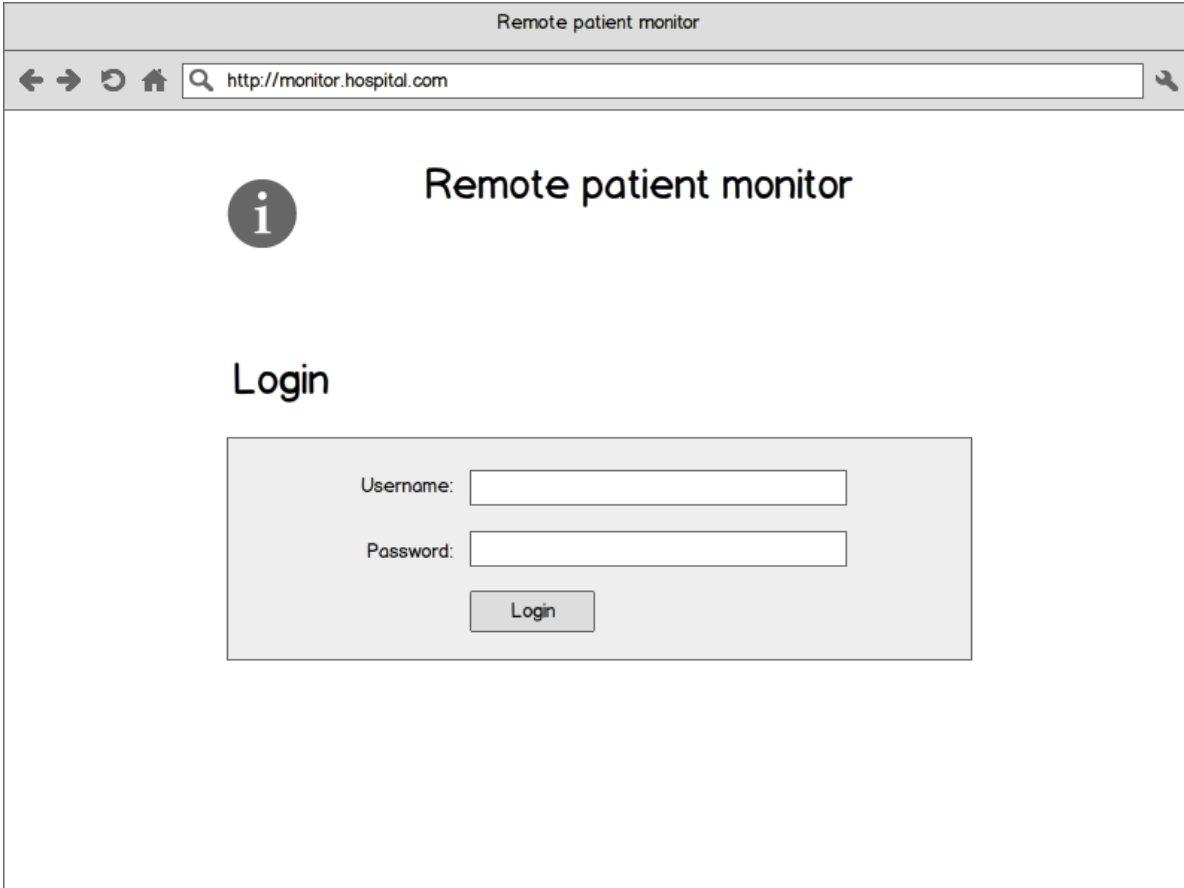
<sup>5</sup> MAC Address Media Access Control - um endereço físico associado à interface de comunicação (MAC Address, 2013)

localization	varchar(255)	Localização do dispositivo
--------------	--------------	----------------------------

Tabela 10 - device\_info (Plataforma WISEcare)

### 3.4.4 Mockups

Nesta parte é apresentada uma modelação da aplicação em versão de *mockups*. Com este tipo de modelação é possível mostrar qual será o aspeto final da aplicação, e assim verificar se todas as funcionalidades anteriormente identificadas estão contempladas.



The mockup shows a web browser window titled "Remote patient monitor". The address bar contains "http://monitor.hospital.com". The main content area features a large circular icon with the letter 'i' on the left and the title "Remote patient monitor" on the right. Below the title is the word "Login". A light gray rectangular box contains the login form, which includes a "Username:" label and a text input field, a "Password:" label and a text input field, and a "Login" button positioned below the password field.

Figura 6 – Login (*mockup*)

No ecrã de login, Figura 6, o utilizador identifica-se no sistema e em seguida entra na aplicação.

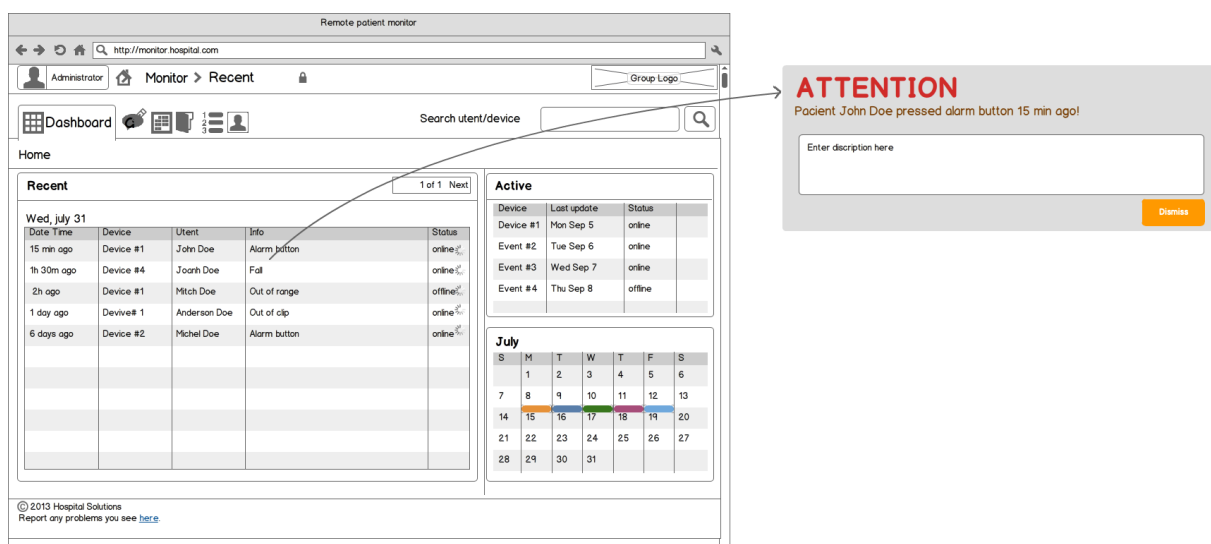


Figura 7 – Dashboard (mockup)

O ecrã exibido na Figura 7 mostra toda a atividade do sistema. Tem uma área com os últimos eventos ocorridos, tem a lista dos dispositivos ativos no sistema e permite navegar para as outras funcionalidades da aplicação. Quando é recebido um evento crítico, surge uma mensagem de alerta ao utilizador do sistema. Ao dispensar a mensagem, é possível inserir um comentário com uma descrição do evento.

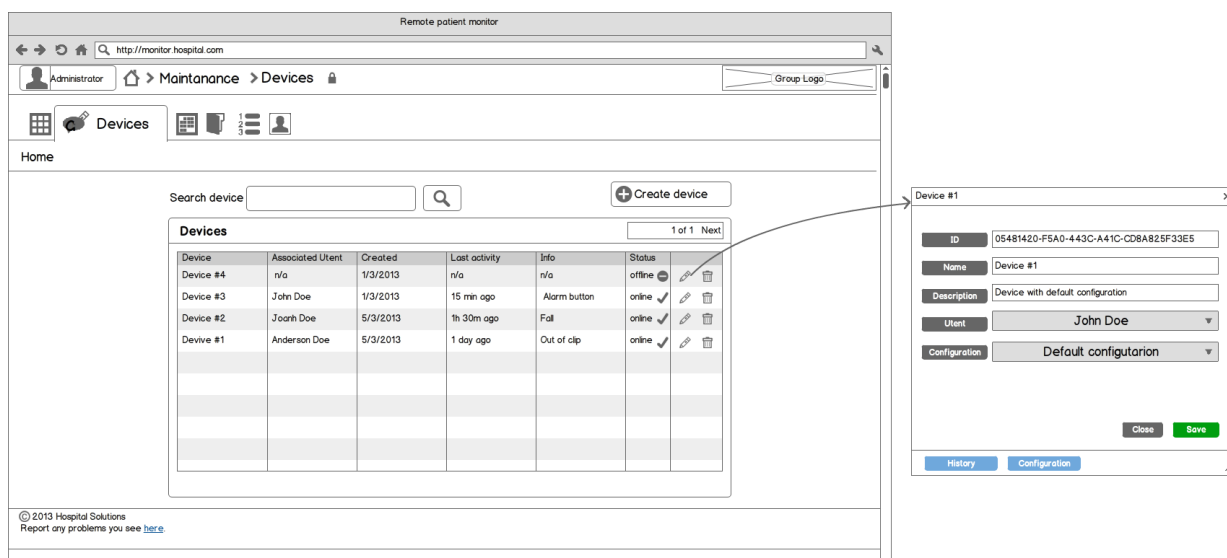


Figura 8 – Dispositivos (mockup)

O ecrã apresentado na Figura 8 serve para fazer a manutenção dos dispositivos, ele disponibiliza as operações “CRUD”, *Create, read, update e delete*.

Foram escolhidos os campos mais relevantes para serem mostrados na grelha.

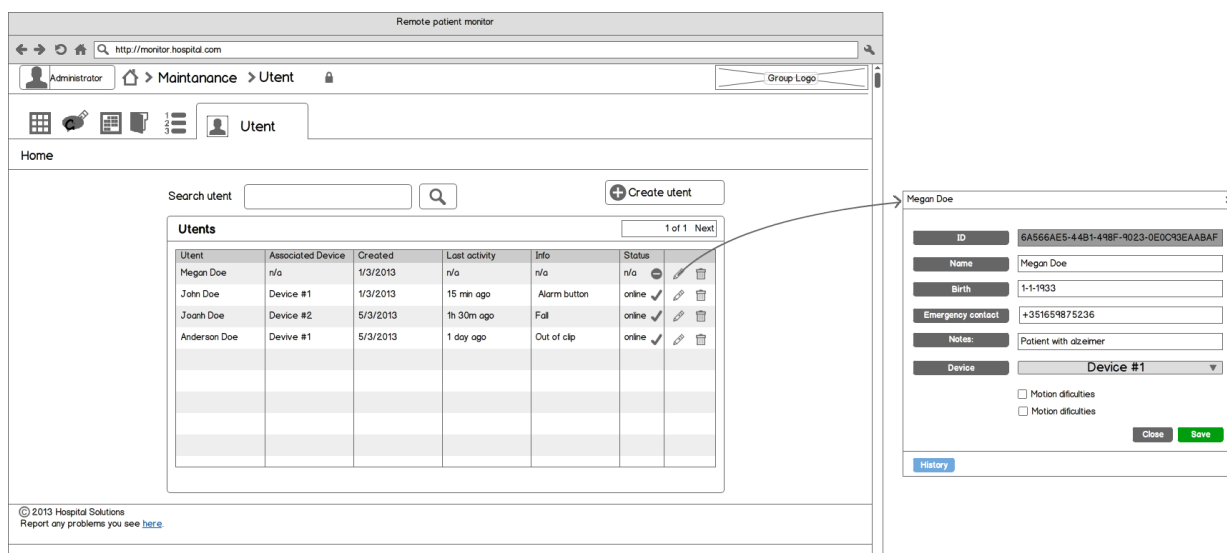


Figura 9 – Utente (mockup)

O ecrã mostrado na Figura 9 serve para fazer a manutenção dos utentes, disponibilizando as operações normais para listar, criar, atualizar e eliminar registos. Os mesmos só podem ser eliminados no caso de não haver outros registos relacionados, uma vez que o SGBD contém as respetivas relações entre as tabelas e garante, dessa forma, a integridade dos dados.

Optou-se por colocar os campos mais importantes visíveis na grelha como o nome do utente, o dispositivo associado, a data da última atividade, informação sobre os alertas e o estado do dispositivo.

No formulário de edição estão disponíveis os campos de dados relativos ao utente e o dispositivo atualmente associado.

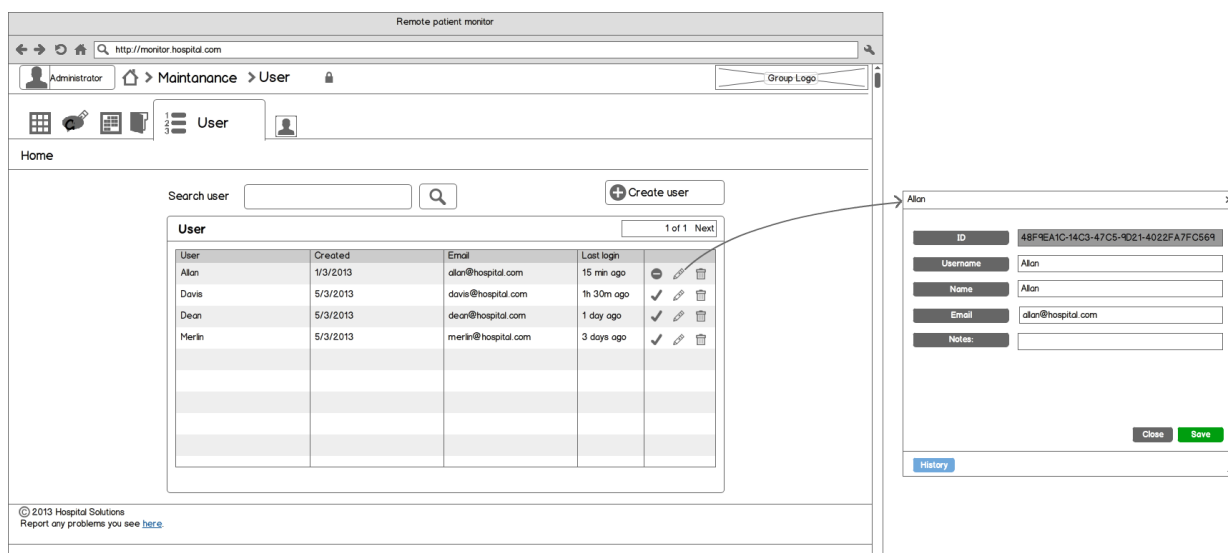


Figura 10 – Utilizadores (mockup)

A opção de utilizadores, Figura 10, mostra os utilizadores registados no sistema, sendo possível criar um novo ou alterar/eliminar um existente. Só é possível eliminar um utilizador se ainda não existir informação associada, isto é, se existirem registos no sistema, já não é possível eliminar, apenas se pode desativá-lo. No entanto continua a existir para efeitos de consulta de histórico.

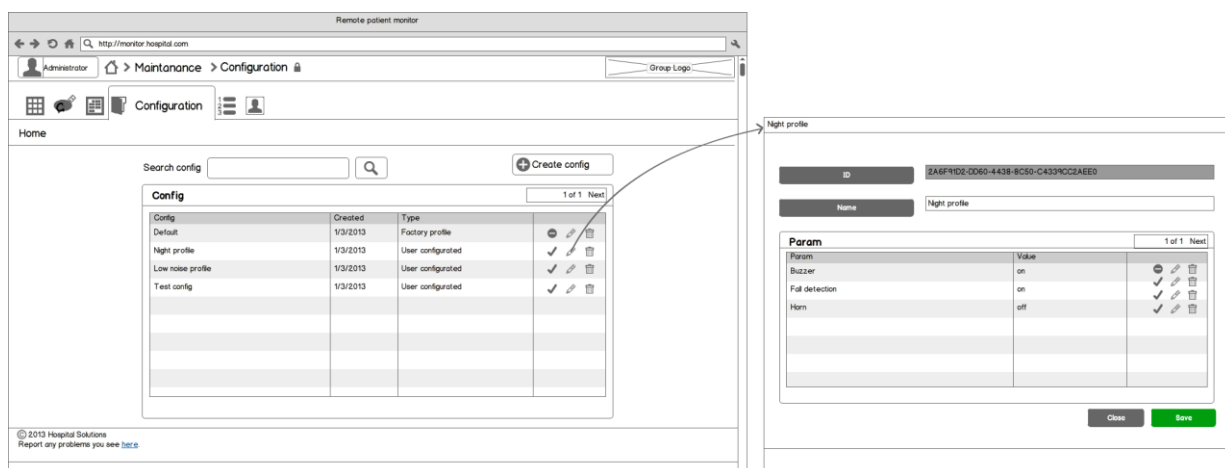


Figura 11 – Configurações (mockup)

A opção de configurações, Figura 11, mostra as configurações de dispositivos que existem no sistema. É possível criar uma nova ou abrir uma configuração e alterar os seus parâmetros. As configurações são aplicadas ao dispositivo quando o mesmo se associa a um utente.



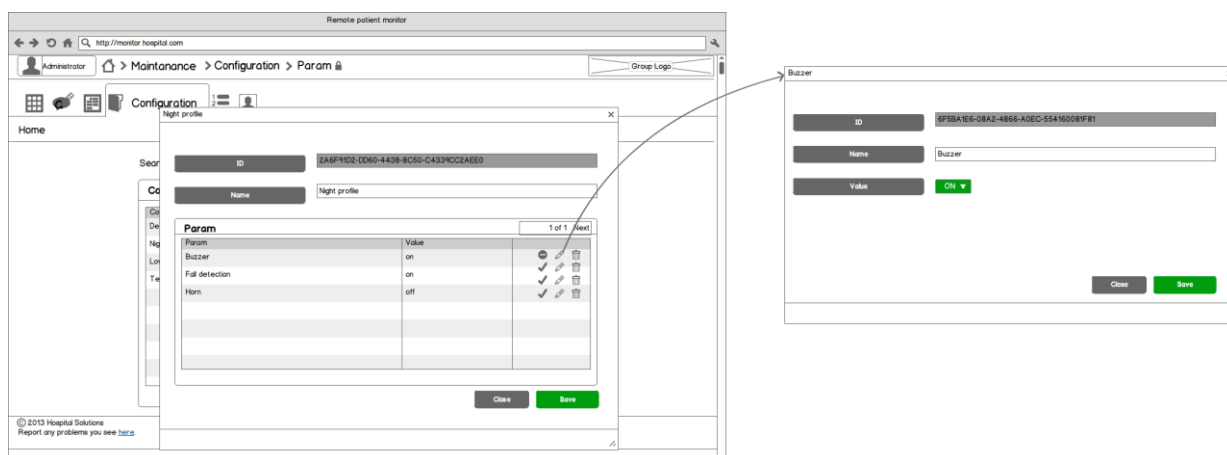


Figura 12 - Parâmetros de configuração (mockup)

O ecrã mostrado na Figura 12 serve para alterar um parâmetro de uma configuração. Os parâmetros são fixos pelo sistema, pelo que não é possível criar novos registos.

Os parâmetros podem ter valores booleanos ou valores discretos e o formulário deve apresentar um controlo diferente consoante o tipo de dados do parâmetro.

### 3.5 Análise da Gateway WISEcare

#### 3.5.1 Modelo de dados (Gateway WISEcare)

A Figura 13 apresenta o modelo de dados e representa as entidades da *Gateway WISEcare*.

São quatro as entidades:

- **Config:** Mantém apenas um registo com a configuração da *gateway*;
- **Device:** Representa um dispositivo e possui campos com o estado atual deste;
- **Params:** Serve para guardar os parâmetros de configuração dos dispositivos, nome do parâmetro e valor;
- **Event:** Guarda os eventos ocorridos no sistema.

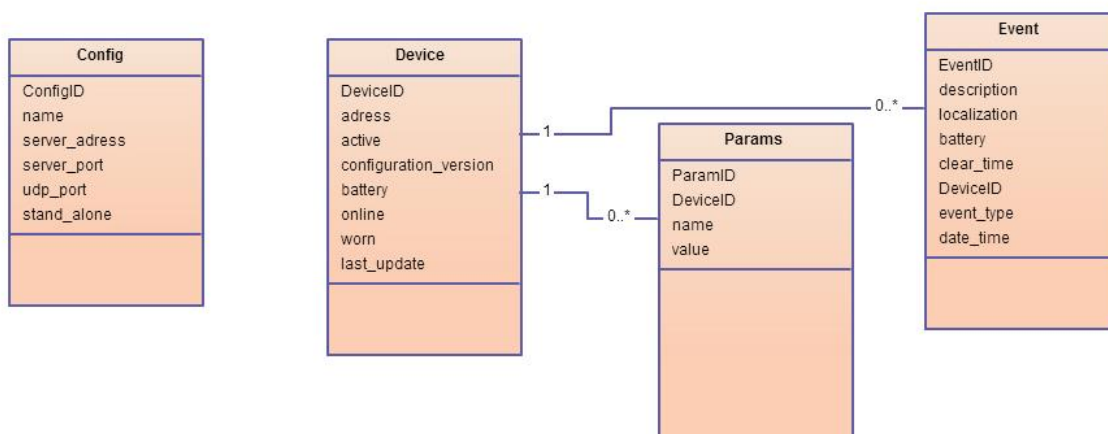


Figura 13 - Modelo de dados (Gateway WISEcare)

### 3.5.2 Modelo físico (Gateway WISEcare)

A Figura 14 é uma representação do modelo físico da base de dados do *Gateway WISEcare* e serviu de base para as tabelas que são apresentadas nos pontos seguintes.

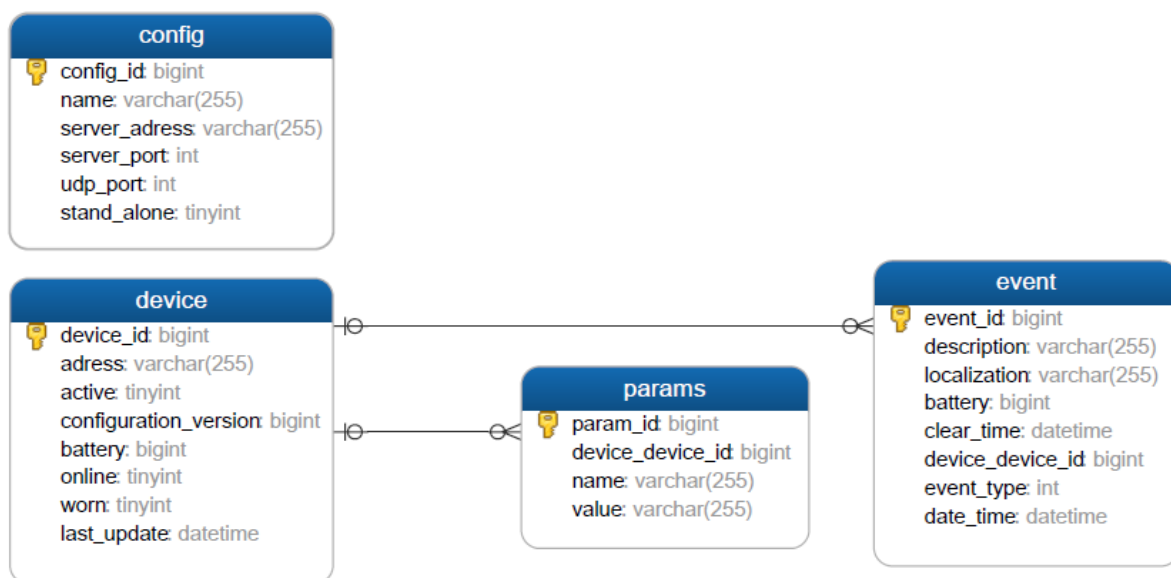


Figura 14 - Modelo físico(Gateway WISEcare)

### 3.5.3 Descrição dos campos do modelo físico (Gateway WISEcare)

Nesta seção do relatório irá proceder-se à descrição das tabelas do modelo físico, campo a campo.

#### 3.5.3.1 Tabela Config

A tabela *Config*, Tabela 11, armazena os dados dos utilizadores do sistema.

Tabela Config		
Nome campo	Tipo campo	Descrição
config_id	bigint	Chave primária
name	varchar(255)	Nome do Gateway
server_adress	varchar(255)	Endereço do servidor onde funciona a plataforma. De onde serão obtidas as funcionalidades quando não funciona como Stand alone. (1)
server_port		Porta do servidor. (1)
udp_port		Porta UDP onde os dispositivos se ligam por udp.
stand_alone		Flag que indica se esta gateway funciona como stand alone ou ligada a uma plataforma.

Tabela 11 – Config (Gateway WISEcare)

- (1) Nota: Não é necessário configurar este endereço se a *gateway* funcionar no modo *stand alone*.

### 3.5.3.2 Tabela device

A tabela *device*, Tabela 12, armazena a informação dos dispositivos registados na *gateway*.

Tabela device		
Nome campo	Tipo campo	Descrição
device_id	bigint	Chave primária
adress	varchar(255)	Endereço MAC
active	tinyint(1)	Indica se o dispositivo se encontra ativo no sistema
configuration_version	bigint	Versão da configuração. Este campo é usado para a sincronização de configurações. Quando a <i>Gateway WISEcare</i> tenta sincronizar as configurações, envia esta versão. Servidor só envia a configuração se for superior.
battery	bigint	Nível atual da bateria
online	tinyint	Informação se o dispositivo se encontra <i>online</i> .
worn	tinyint	Informação se o dispositivo está a ser usado ou fora do clip
last_update	datetime	Última atualização/comunicação

Tabela 12 - Tabela device (*Gateway WISEcare*)

### 3.5.3.3 Tabela event

A tabela *event*, Tabela 13, guarda os vários eventos que provêm dos dispositivos ligados ao sistema.

Tabela event		
Nome campo	Tipo campo	Descrição
event_id	bigint	Chave primária
description	varchar(255)	Nome da configuração
localization	varchar(255)	Descrição da configuração
battery	bigint	Estado da bateria do dispositivo
clear_time	datetime	Data/Hora que o evento foi limpo
device_device_id	bigint	Chave estrangeira da tabela <i>device</i>
event_type	int	Tipo de evento
date_time	datetime	Data/Hora que o evento surgiu

Tabela 13 - Tabela event (*Gateway WISEcare*)

### 3.5.3.4 Tabela params

A tabela *params*, Tabela 14, guarda os valores dos parâmetros de um dispositivo.

Tabela configuration_params		
Nome campo	Tipo campo	Descrição
param_id	bigint	Chave primária
device_device_id	bigint	Chave estrangeira tabela <i>device</i>
name	varchar(255)	Nome do parâmetro
value	varchar(255)	Valor do parâmetro

Tabela 14 - Tabela configuration\_params (*Gateway WISEcare*)

## Capítulo 4. Implementação do projeto

### 4.1 Recursos usados para implementação

#### 4.1.1 Modelo de implementação – MVC

Tendo em conta que o projeto foi implementado usando a plataforma Play, torna-se praticamente incontornável a utilização do padrão de desenvolvimento Model-View-Controller (MVC).

MVC é um modelo de desenvolvimento de *Software*, atualmente considerado um Design Pattern (padrão de projeto) utilizado na Engenharia de Software. O modelo isola a "lógica" (a lógica da aplicação) da interface do usuário (inserir e exibir dados), permitindo desenvolver, editar e testar separadamente cada parte (Padrão MVC - Wikipédia, 2013).

#### 4.1.2 Modelo - Acesso a dados através do EBEAN

Object Relational mapping (ORM) é uma técnica de programação para converter tipos de dados que não são compatíveis em linguagens orientadas a objetos. Cria efetivamente uma base de dados virtual que pode ser usada na linguagem de programação (ORM - Wikipedia, 2013).

O EBEAN é um modelo de mapeamento de objetos relacional (ORM) para java. Foi desenhado para ser simples de usar e compreender (Ebean Wikipedia, 2012).

Como a plataforma usa o EBEAN ORM, basta simplesmente criar o conjunto de classes que representam o modelo de dados. Toda a complexidade de ligação à base de dados, consulta, inserção e atualização de dados é encapsulada na camada de acesso que é fornecida pelo EBEAN.

Recorrendo ao Eclipse foi criado o modelo de dados de acordo com a especificação inicial.

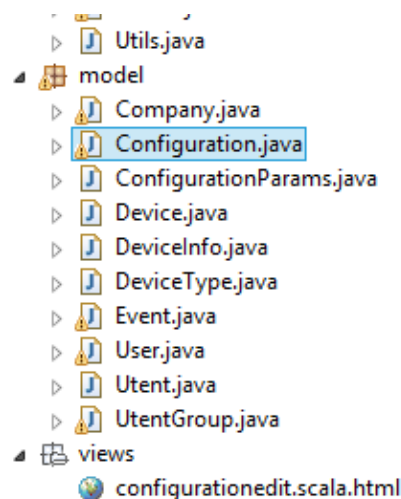


Figura 15 - Modelo



### 4.1.3 Definição das Vistas

A plataforma play vem com um motor muito poderoso baseado em *templates Scala*. Estes *templates* são essencialmente:

- Compactos, expressivos, e fluidos: minimizam o código que é necessário em cada ficheiro e permitem um processo de desenvolvimento mais rápido e intuitivo;
- De fácil aprendizagem: permitem que se usem os conhecimentos básicos de *html* e/ou *scala*;
- Editáveis em qualquer editor de texto, não requerem ferramentas específicas de edição;
- Compilados na aplicação, pelo que permitem detetar imediatamente os erros e visualizá-los diretamente no *browser* (Templates Scala, 2013).

Nesse sentido, foi necessário definir as várias vistas (*templates*), Figura 16, para cada funcionalidade a implementar na plataforma.

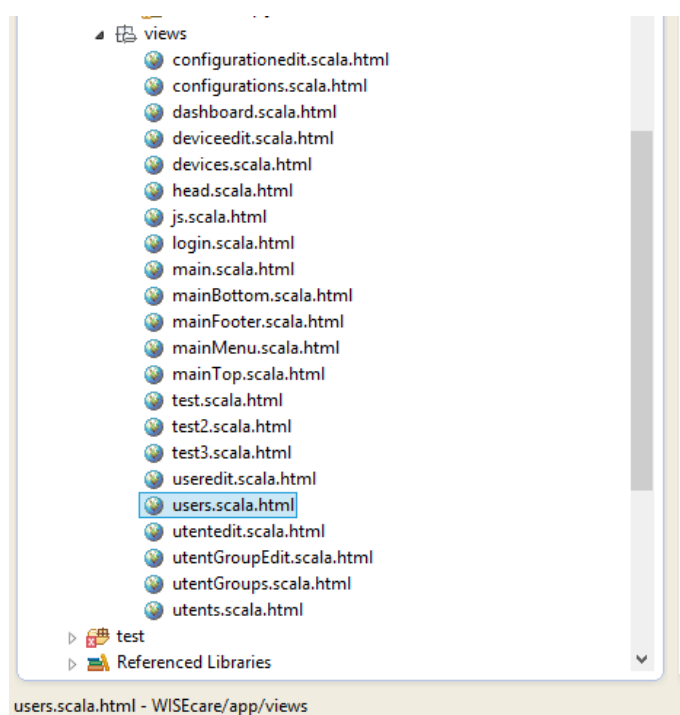


Figura 16 - Vistas

## 4.1.4 Sistema de notificações

A característica fundamental deste projeto é a capacidade de alerta dos monitores para as situações de emergência. Assim, neste projeto foi implementado um sistema de notificações para que os utilizadores (monitores) recebam em tempo real as notificações que ocorrem no sistema. Essas notificações são normalmente originadas pelos dispositivos usados pelos utentes e ligados à plataforma via *wireless*.

Como a plataforma funciona em ambiente web, recorreu-se à utilização de *websockets* para enviar as notificações para os clientes (*browser*).

*WebSocket* é uma tecnologia que permite a comunicação bidirecional por canais *full-duplex* sobre um único soquete *Transmission Control Protocol* (TCP). Ele é projetado para ser executado em *browsers* e servidores web que suportem o HTML5, mas pode ser usado por qualquer cliente ou servidor de aplicativos. A API *WebSocket2* está a ser padronizada pelo W3C3 e o protocolo *WebSocket* está a ser padronizado pelo IETF4 (*WebSockets* -Wikipedia ).

#### 4.1.5 Interface

Relativamente ao modo de funcionamento de pesquisa e criação e edição de dados, um dos requisitos do projeto era a sua simplicidade para o utilizador. Assim, durante a fase de análise e implementação houve sempre o cuidado de manter as funcionalidades disponíveis da forma mais simples possível, mantendo o mesmo aspeto entre os vários ecrãs. As opções mais importantes também foram colocadas de forma bem visível para o utilizador. Como as plataformas funcionam em ambiente web, prevê-se que a maioria dos utilizadores já esteja familiarizada com os vários controlos disponíveis.

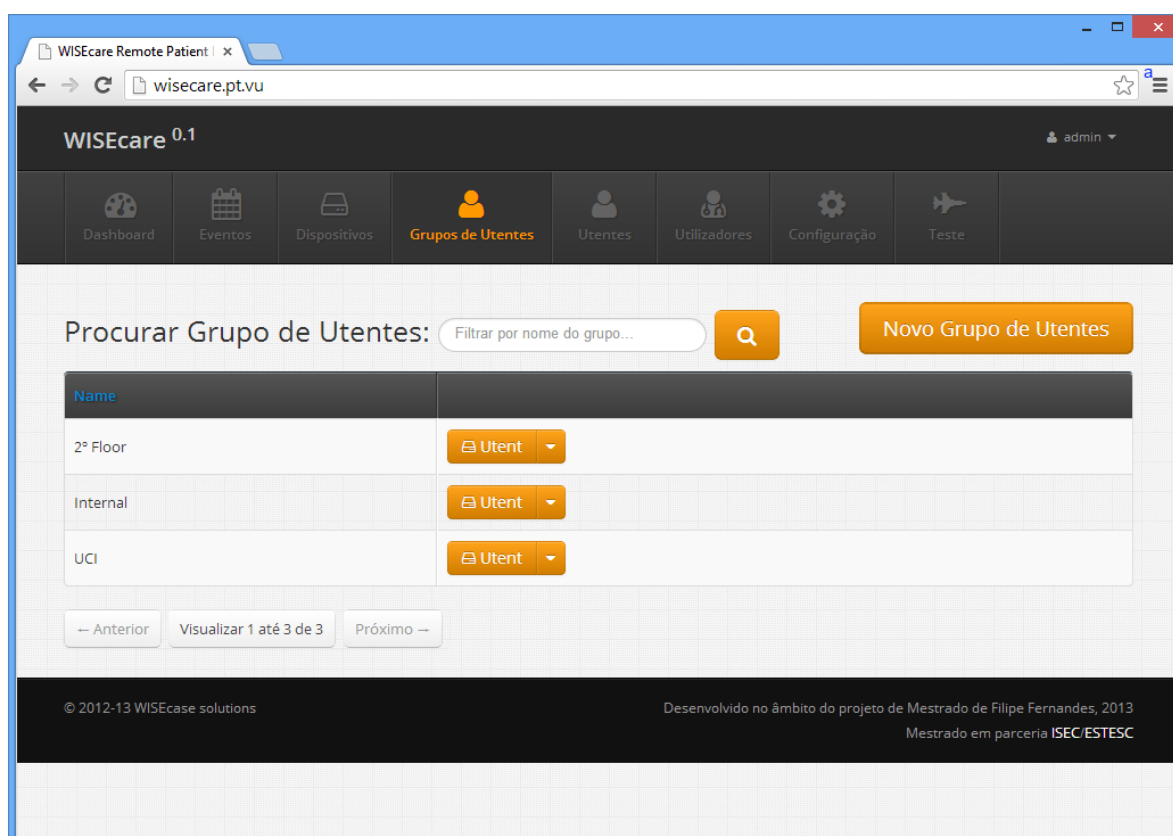


Figura 17 - Exemplo de navegação

O método de navegação tanto no portal como no *gateway* é sempre igual. Para executar a maioria das funcionalidades (veja-se um exemplo na Figura 17), o utilizador acede à opção correspondente através do menu superior e em seguida é apresentada, em forma de grelha, a primeira página com os registos existentes. Desta forma, o utilizador tem sempre os registos mais recentes em primeiro lugar. Existe sempre uma caixa de pesquisa para procurar um registo em particular e botões para navegar para as páginas seguintes. Desta forma garante-se que a página carrega sempre de forma rápida para o utilizador. No caso

de existirem operações a fazer sobre um item, a opção encontra-se no *drop-down* que aparece na grelha.

O botão para criar um novo registo também está bem visível logo na parte superior da grelha.

## 4.1.6 Implementação da *Plataforma WISEcare*

### 4.1.6.1 Autenticação

Como este sistema armazena informação confidencial de pessoas, existe um sistema de autenticação que apenas permite que utilizadores devidamente autenticados visualizem os dados, Figura 18.

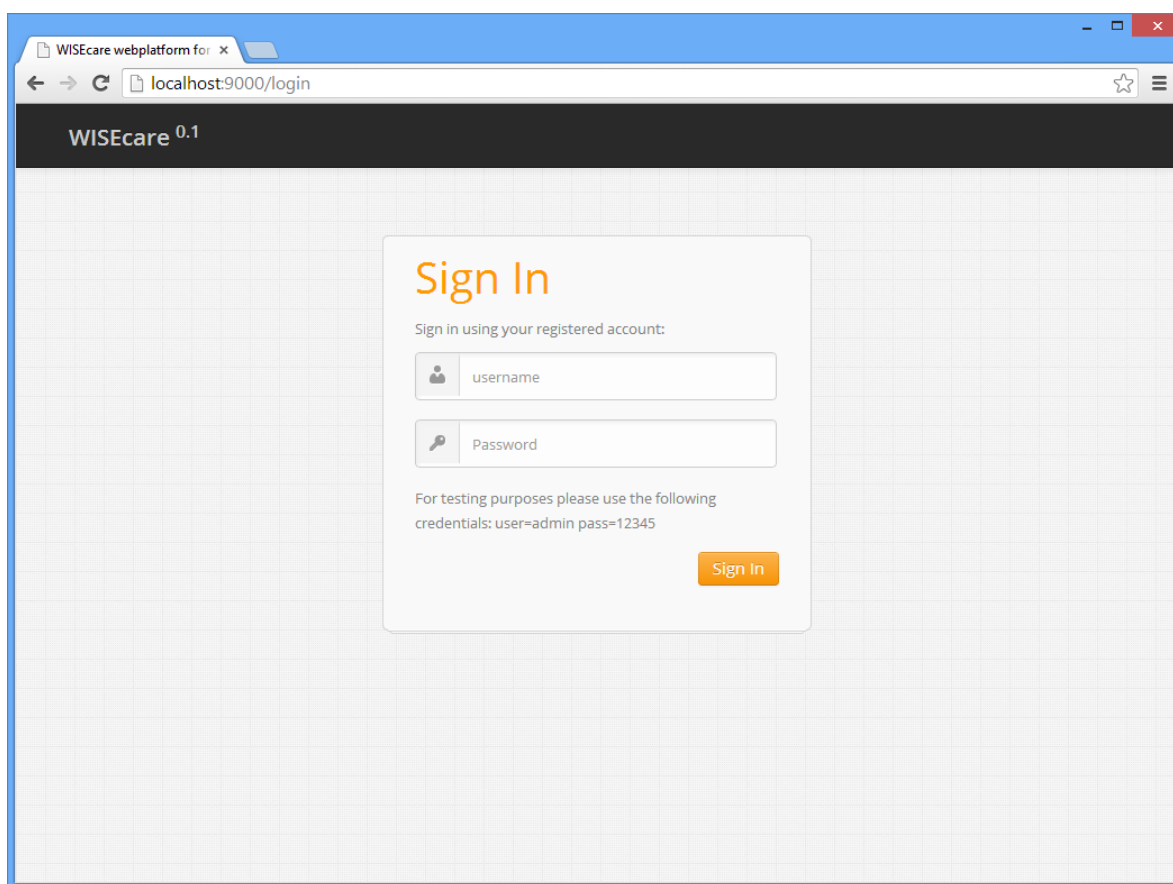


Figura 18 - Login

#### 4.1.6.2 Dashboard

No *dashboard*, Figura 19, o utilizador pode ter uma vista geral do estado dos pacientes que estão a ser monitorizados por si. Se existir um alarme, surge um *popup* com os dados do utente e informação sobre a ocorrência. O *dashboard* mostra informações novas em tempo real, à medida que acontecem.

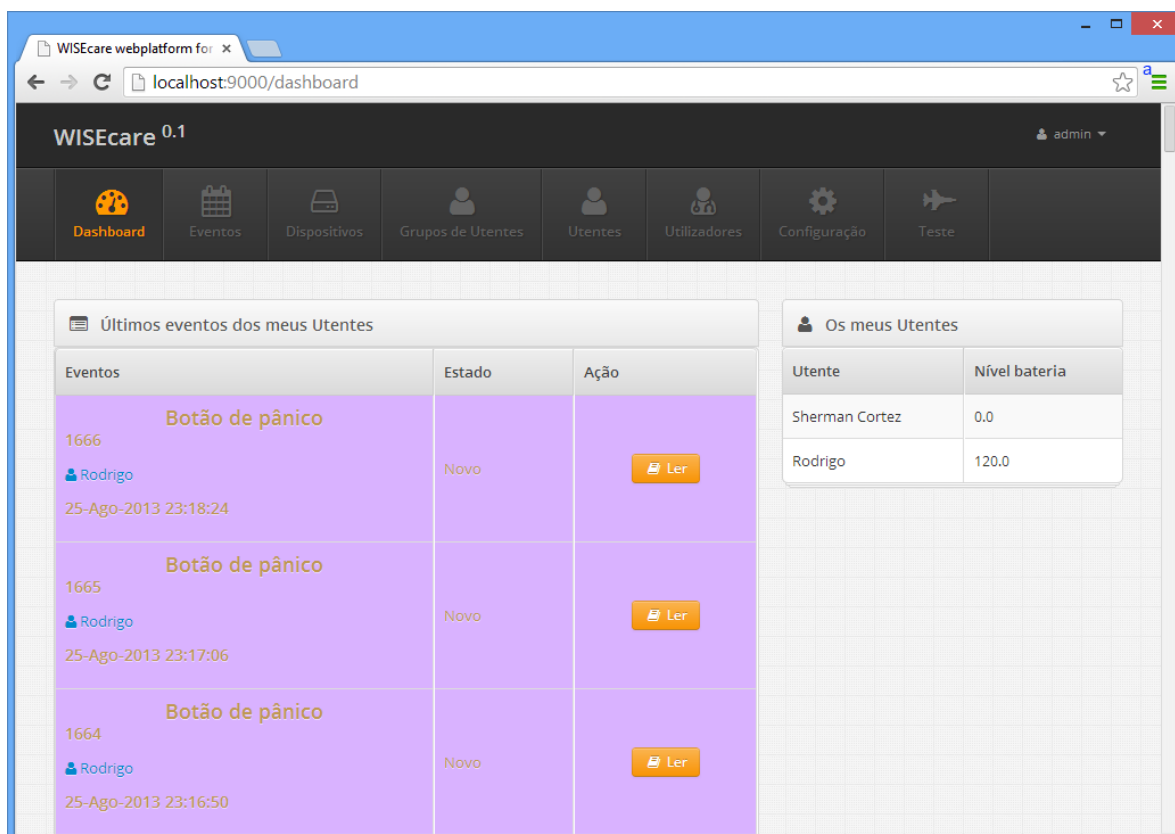


Figura 19 - Dashboard

## 4.1.6.3 Dashboard – Evento

Após visualizar um evento, é possível inserir uma descrição da ação que foi tomada na sequência do evento, Figura 20.

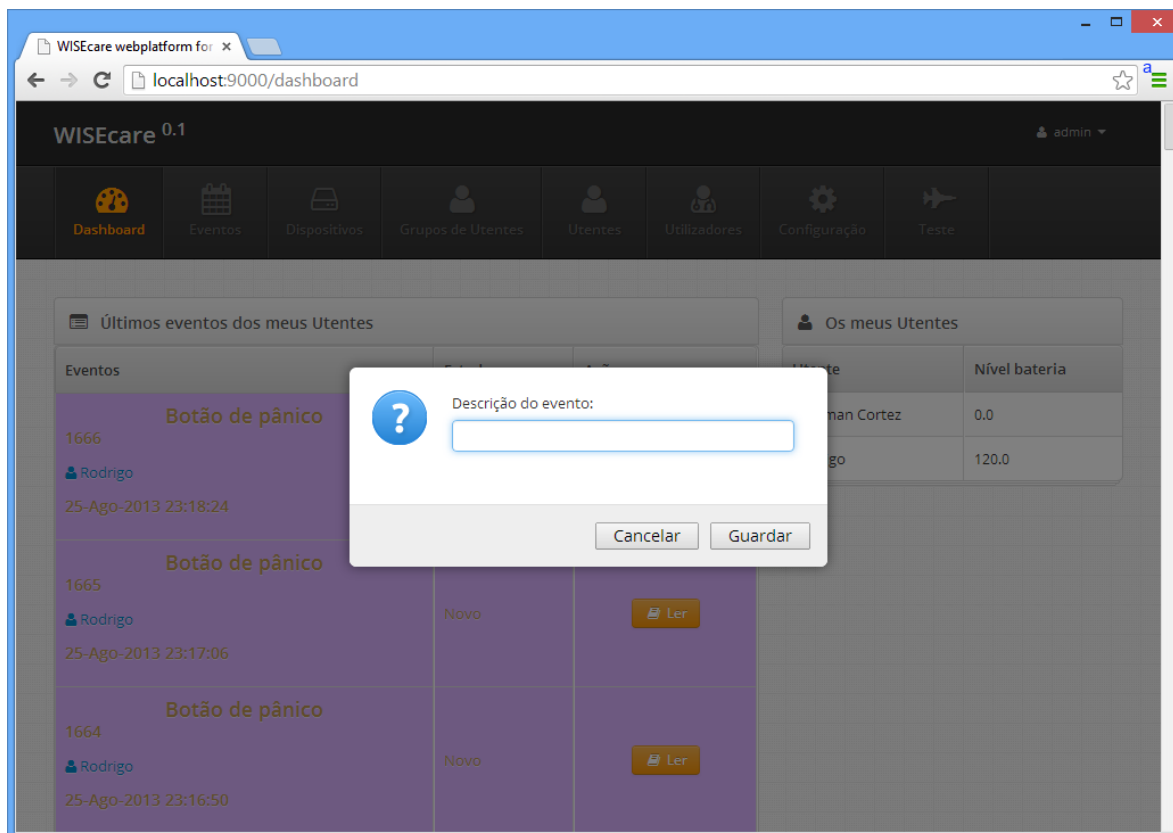
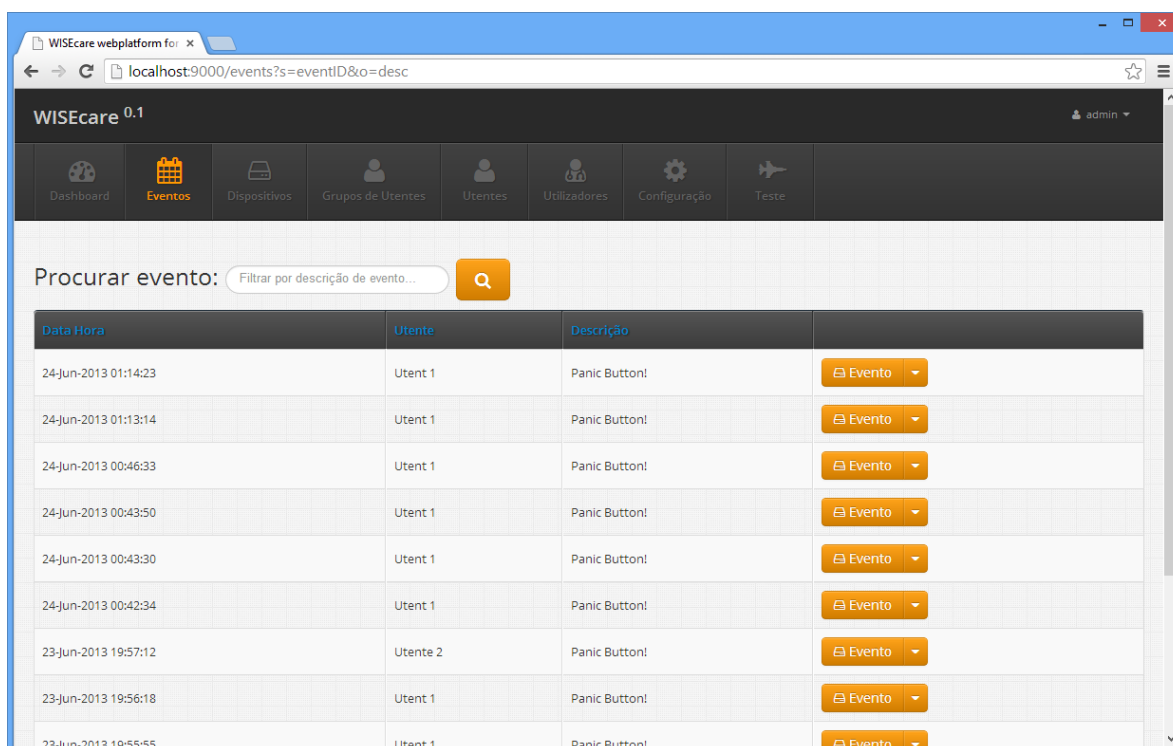


Figura 20 Dashboard – Evento

## 4.1.6.4 Eventos

No ecrã de eventos, Figura 21, é possível visualizar o registo de todos os eventos ocorridos no sistema. É possível ordenar por data/hora do evento, nome do utente ou descrição do evento. Também é possível filtrar por descrição de evento.

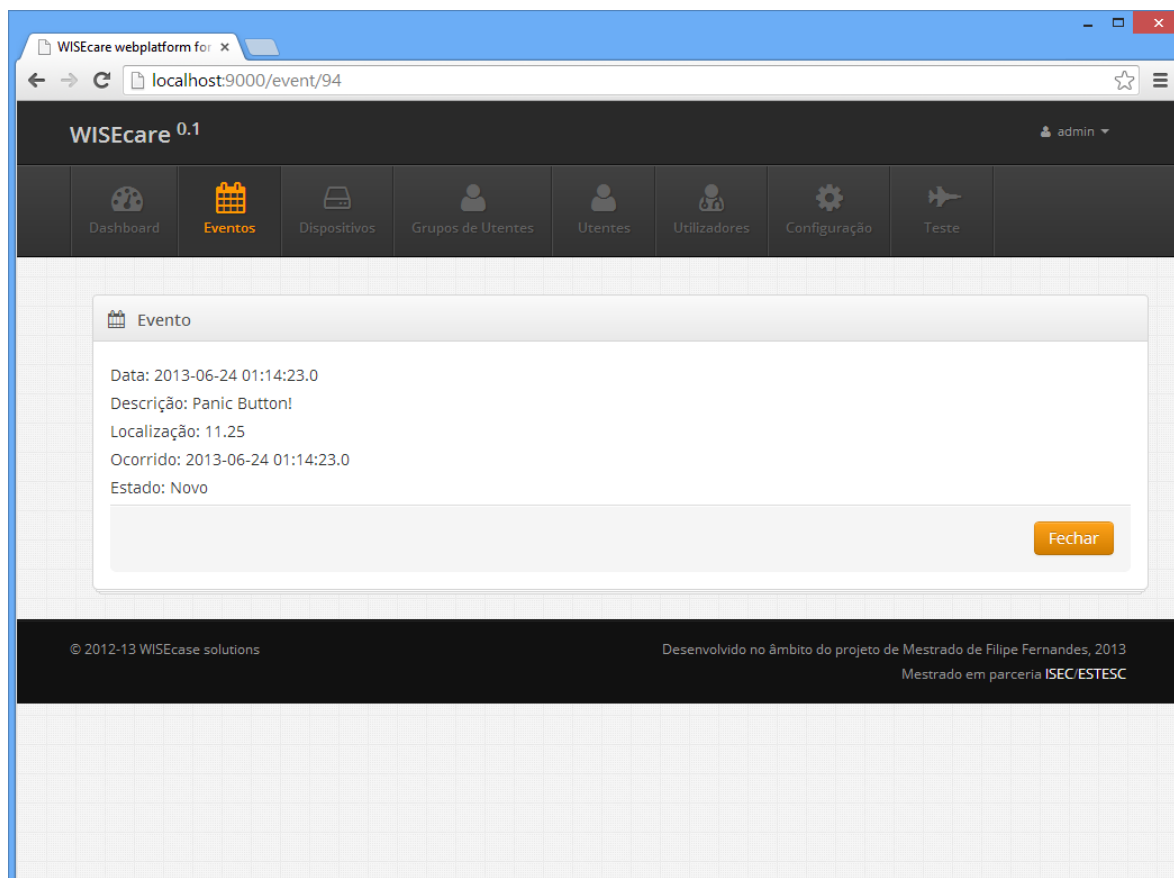


Data Hora	Utente	Descrição	Evento
24-Jun-2013 01:14:23	Utent 1	Panic Button!	Evento
24-Jun-2013 01:13:14	Utent 1	Panic Button!	Evento
24-Jun-2013 00:46:33	Utent 1	Panic Button!	Evento
24-Jun-2013 00:43:50	Utent 1	Panic Button!	Evento
24-Jun-2013 00:43:30	Utent 1	Panic Button!	Evento
24-Jun-2013 00:42:34	Utent 1	Panic Button!	Evento
23-Jun-2013 19:57:12	Utente 2	Panic Button!	Evento
23-Jun-2013 19:56:18	Utent 1	Panic Button!	Evento
23-Jun-2013 19:55:55	Utent 1	Panic Button!	Evento

Figura 21 - Eventos



Permite também visualizar os detalhes do evento, a data/hora de ocorrência, o nome do utente, a localização, a data/hora de visualização e descrição que o monitor colocou quando visualizou o evento, conforme ilustrado na imagem da Figura 22.



**Figura 22 - Eventos - visualizar**

#### 4.1.6.5 Dispositivos

No ecrã apresentado na Figura 23 é possível fazer a manutenção dos dispositivos do sistema, além de mostrar os existentes, com opção de ordenação e filtragem. Dispõe das seguintes funcionalidades: criar, editar ou eliminar um dispositivo.

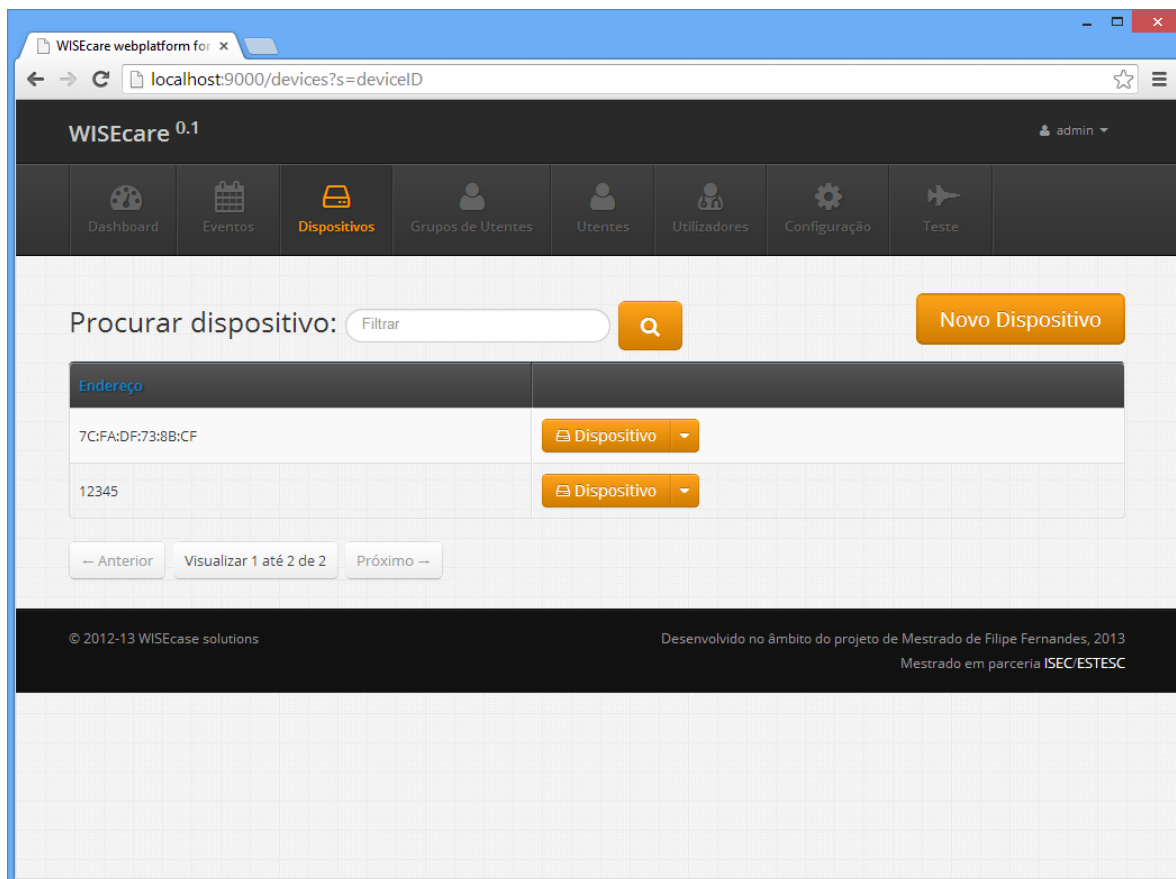


Figura 23 – Dispositivos

No ecrã de dispositivos, Figura 24, regista-se apenas o *mac-adress* do dispositivo e configura-se a *flag* “ativo”. Apenas os dispositivos ativos podem ser associados a utentes. Se um dispositivo avariar ou deixar de existir, deve-se desativar a *flag* “ativo”.

The screenshot shows a web browser window with the URL `localhost:9000/device/1`. The page title is "WISEcare 0.1" and the user is logged in as "admin". The navigation menu includes: Dashboard, Eventos, Dispositivos (highlighted), Grupos de Utentes, Utentes, Utilizadores, Configuração, and Teste. The main content area displays a form titled "Editar Dispositivo 1". The form has a label "Endereço" with a text input field containing "7C:FA:DF:73:8B:CF". Below this is a "Required" label. The "Ativo" section has a checked checkbox. At the bottom right of the form are "Guardar" and "Cancelar" buttons. The footer contains copyright information: "© 2012-13 WISEcase solutions" and "Desenvolvido no âmbito do projeto de Mestrado de Filipe Fernandes, 2013 Mestrado em parceria ISEC/ESTESC".

Figura 24 - Dispositivos - Editar

#### 4.1.6.6 Grupos de utentes

É possível organizar utentes por grupos, Figura 25. A partir do ecrã da Figura 26 criam-se os diversos grupos que vão existir no sistema.

Os grupos de utentes são de extrema importância num cenário em que a aplicação é usada num hospital com muitos utentes. Ao criar grupos e organizar os utentes nos respetivos grupos, torna-se mais fácil associar/desassociar os utentes aos monitores.

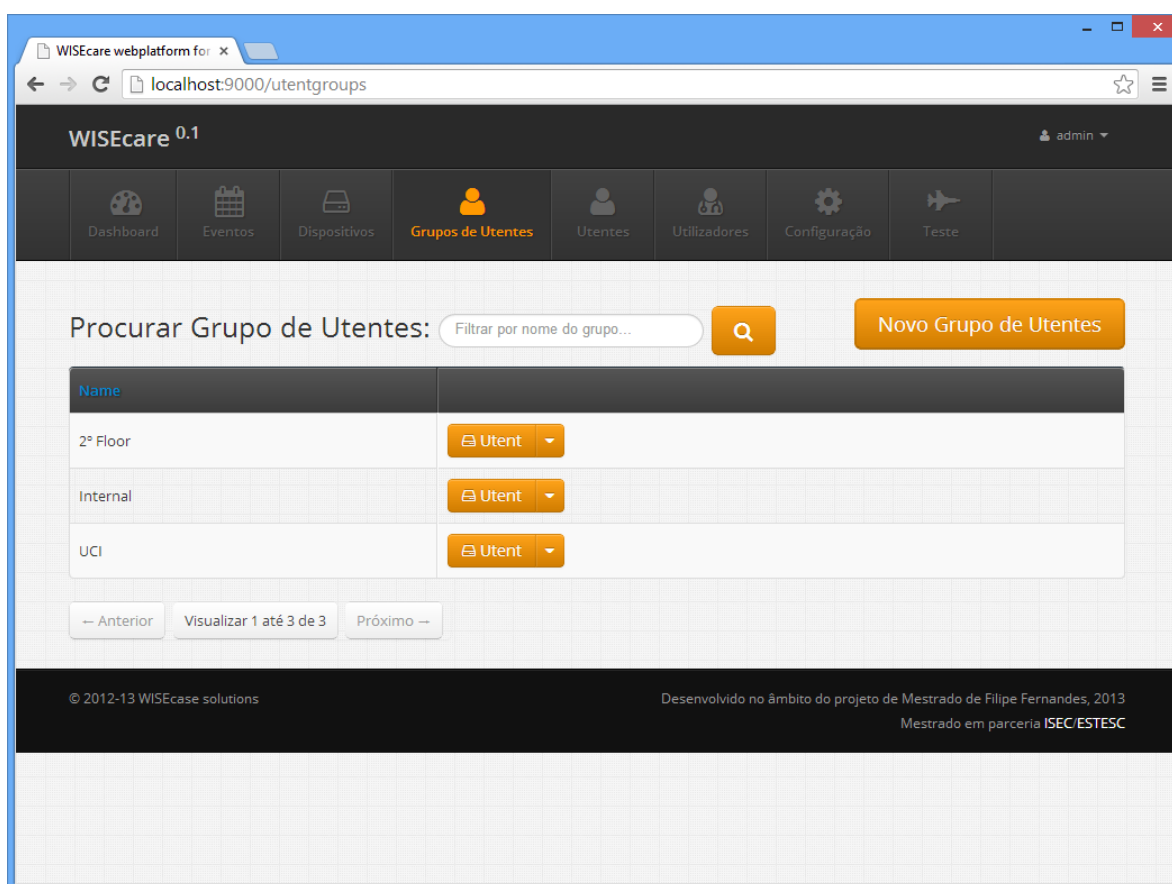


Figura 25 - Grupos de utentes

The screenshot displays a web browser window with the address bar showing 'localhost:9000/newUtentGroup'. The page title is 'WISEcare 0.1' and the user is logged in as 'admin'. The navigation menu includes: Dashboard, Eventos, Dispositivos, Grupos de Utentes (highlighted), Utentes, Utilizadores, Configuração, and Teste. The main content area features a form titled 'Novo Grupo de Utentes' with a 'Nome' input field and a 'Required' label. At the bottom of the form are 'Guardar' and 'Cancelar' buttons. The footer contains copyright information: '© 2012-13 WISEcase solutions' and 'Desenvolvido no âmbito do projeto de Mestrado de Filipe Fernandes, 2013 Mestrado em parceria ISEC/ESTESC'.

Figura 26 - Grupos de utentes – Novo/Editar

#### 4.1.6.7 Utentes

No ecrã de manutenção de utentes, Figura 27, tal como nos outros ecrãs estão disponíveis funcionalidades de pesquisa, ordenamento e edição.

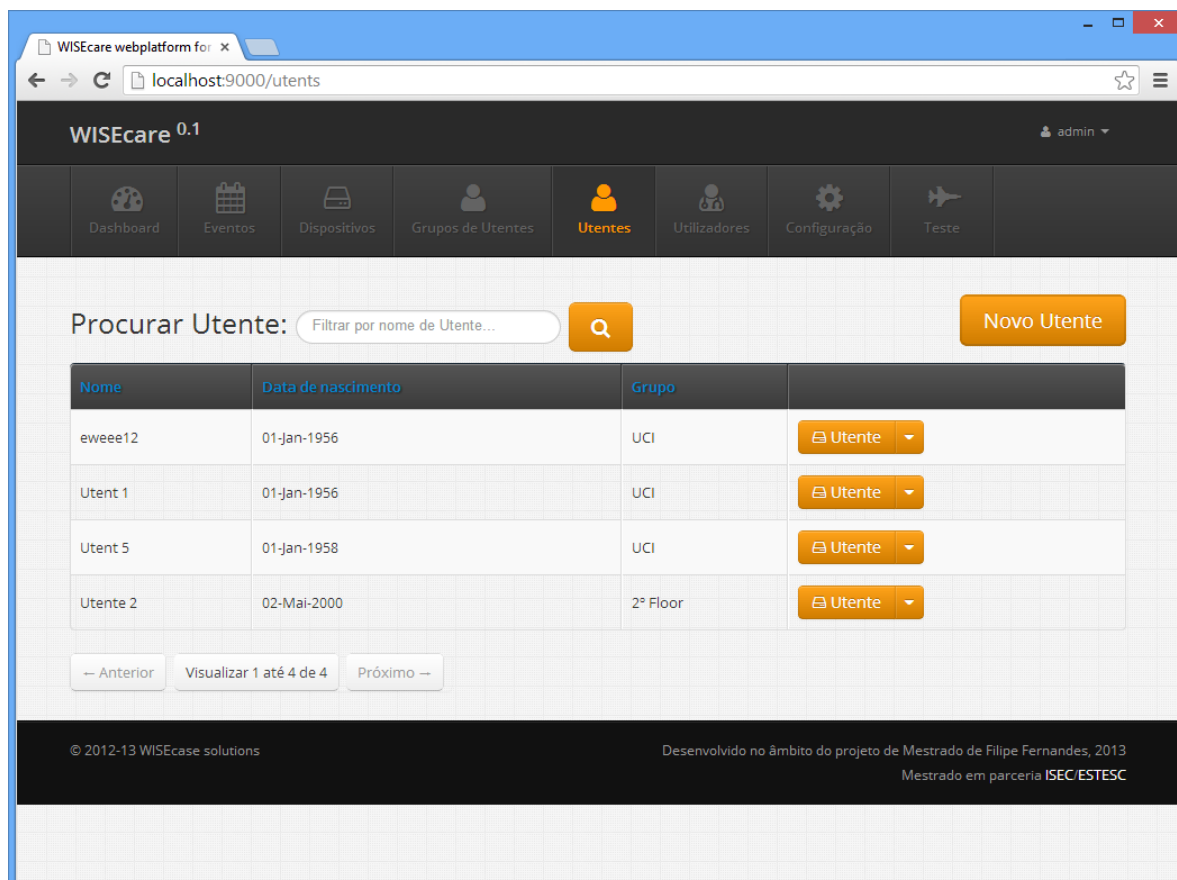
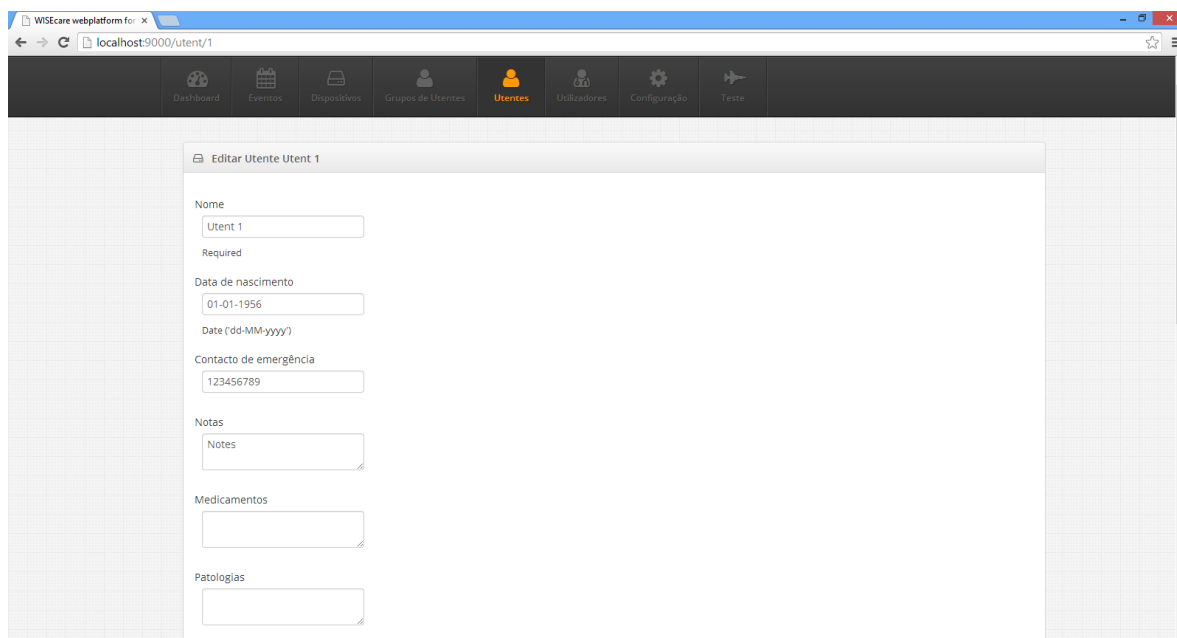


Figura 27 – Utentes

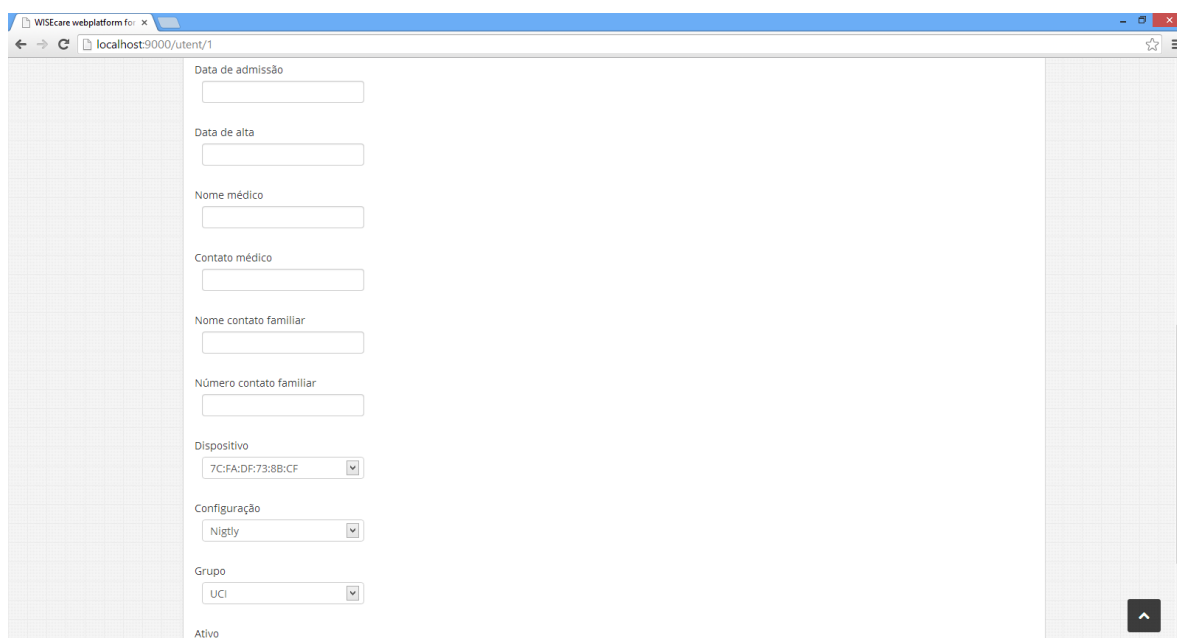
A ficha de utente, Figura 28, Figura 29 e Figura 30, contém as informações relacionadas com o utente, tais como o nome, data de nascimento, contatos dos familiares e do médico, informação sobre medicamentos atualmente a serem tomados pelo utente e patologias diagnosticadas ao utente.



The screenshot shows a web browser window with the URL 'localhost:9000/utente/1'. The browser's address bar and tabs are visible. The page has a dark navigation bar with icons for Dashboard, Eventos, Dispositivos, Grupos de Utentes, Utentes (highlighted), Utilizadores, Configuração, and Teste. The main content area displays a form titled 'Editar Utente Utente 1'. The form contains the following fields: 'Nome' (text input with 'Utente 1'), 'Required' (checkbox), 'Data de nascimento' (date input with '01-01-1956'), 'Date (dd-MM-yyyy)' (text label), 'Contacto de emergência' (text input with '123456789'), 'Notas' (text area with 'Notes'), 'Medicamentos' (text area), and 'Patologias' (text area).

**Figura 28 - Utente – Editar 1/3**

Para além da informação atrás mencionada, ainda se pode encontrar na ficha do utente informação sobre o dispositivo atualmente associada ao mesmo, bem como a configuração que lhe está associada.



The screenshot shows the same web browser window as Figure 28, but the form is scrolled down to show additional fields. The fields include: 'Data de admissão' (date input), 'Data de alta' (date input), 'Nome médico' (text input), 'Contacto médico' (text input), 'Nome contato familiar' (text input), 'Número contato familiar' (text input), 'Dispositivo' (dropdown menu with '7C:FA:DF:73:8B:CF' selected), 'Configuração' (dropdown menu with 'Nigty' selected), 'Grupo' (dropdown menu with 'UCI' selected), and 'Ativo' (checkbox).

**Figura 29 - Utente – Editar 2/3**

Finalmente tem a *flag* “ativo” e o grupo a que pertence.

The screenshot shows a web browser window with the address bar displaying 'localhost:9000/utente/1'. The page contains a form for editing a user profile. The form fields are as follows:

- Contato médico: Text input field.
- Nome contato familiar: Text input field.
- Número contato familiar: Text input field.
- Dispositivo: Dropdown menu with the value '7C:FA:DF:73:8B:CF' selected.
- Configuração: Dropdown menu with the value 'Nightly' selected.
- Grupo: Dropdown menu with the value 'UCI' selected.
- Ativo: A checked checkbox.

At the bottom right of the form, there are two buttons: 'Guardar' (Save) and 'Cancelar' (Cancel). The footer of the page contains the following text:

© 2012-13 WISEcare solutions  
Desenvolvido no âmbito do projeto de Mestrado de Filipe Fernandes, 2013  
Mestrado em parceria ISEC-ESTESC

Figura 30 - Utente – Editar 3/3



## 4.1.6.8 Utilizadores

Existem 2 tipos de utilizadores:

- **Administrador** – Tem acesso a todas as funcionalidades da plataforma;
- **Monitor** – Tem acesso a todas as funcionalidades, exceto aceder ao menu de utilizadores.

No ecrã de utilizadores, Figura 31, estão disponíveis as diversas funcionalidades de manutenção dos utilizadores do sistema. Este ecrã está apenas disponível para utilizadores com privilégios de administração.

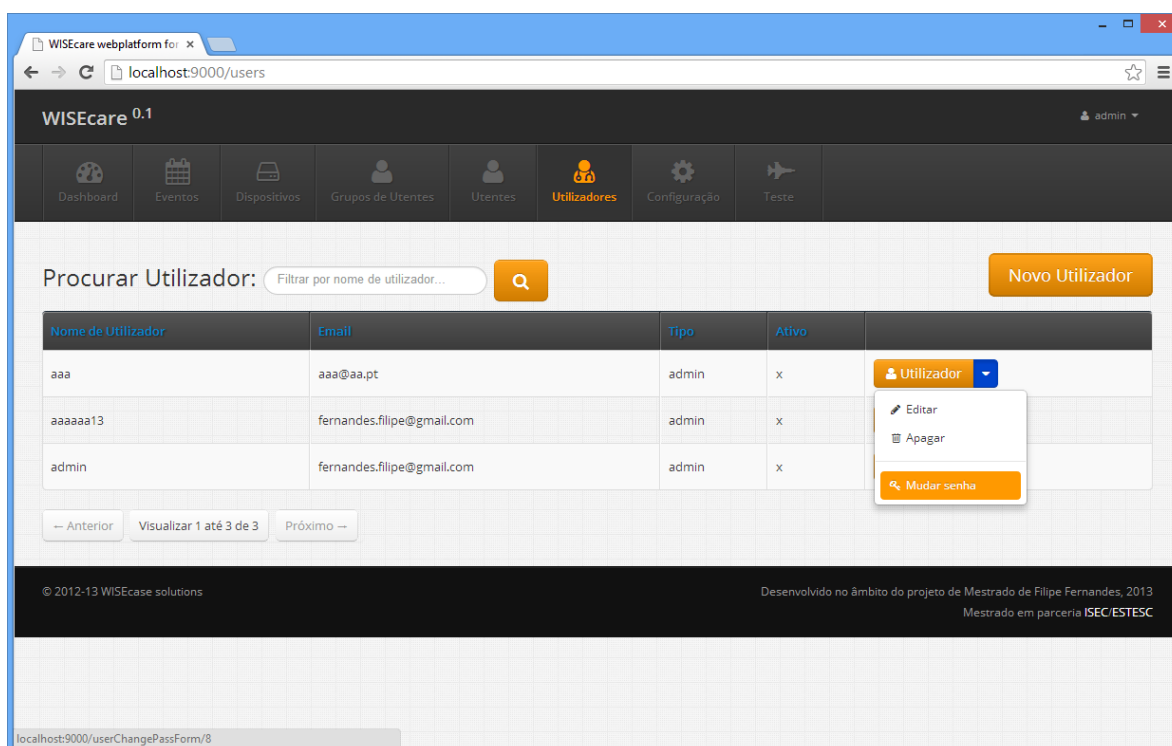


Figura 31 – Utilizadores

No ecrã de edição de utilizador, Figura 32, é possível configurar os dados relativos ao utilizador, como o nome, correio eletrónico e ainda quais os utentes que são monitorizados pelo utilizador.

O *dashboard* apenas vai apresentar notificações dos utentes associados ao utilizador que se autenticou no sistema.

The screenshot shows a web browser window with the address bar displaying 'localhost:9000/user/5'. The browser's tab is labeled 'WISEcare webplatform for x'. The application's navigation bar includes icons and labels for 'Dashboard', 'Eventos', 'Dispositivos', 'Grupos de Utentes', 'Utentes', 'Utilizadores' (highlighted in orange), 'Configuração', and 'Teste'. The main content area is titled 'Editar Utilizador admin' and contains the following form fields:

- Nome de Utilizador:** A text input field containing 'admin'.
- Obrigatório:** A label indicating a required field.
- Email:** A text input field containing 'fernandes.filipe@gmail.com', with an 'Obrigatório' label.
- Primeiro nome:** A text input field containing 'Filipe', with an 'Obrigatório' label.
- Último Nome:** A text input field containing 'Fernandes', with an 'Obrigatório' label.
- 4 selected:** A dropdown menu showing '4 selected'.
- Ativo:** A checkbox that is checked.
- Tipo de utilizador:** A dropdown menu with 'Administrator' selected.

At the bottom right of the form are two buttons: 'Guardar' (grey) and 'Cancelar' (orange). The footer of the browser window shows '© 2012-13 WISEcare solutions' and 'Desenvolvido no âmbito do projeto de Mestrado de Filipe Fernandes, 2012'.

Figura 32 - Utilizadores - Editar

#### 4.1.6.9 Configurações

As configurações, Figura 33, são apenas perfis de configuração dos dispositivos. Cada dispositivo necessita de um conjunto de parâmetros configurados, que são de vários tipos (*flags*, valores discretos), podendo variar entre escalas diferentes e assumir valores diferentes entre si. Para ser mais simples de configurar um dispositivo, cada configuração já contém todos os parâmetros configurados. Na ficha de utente apenas se escolhe qual a configuração pretendida e todos os parâmetros dessa configuração são passados para o dispositivo.

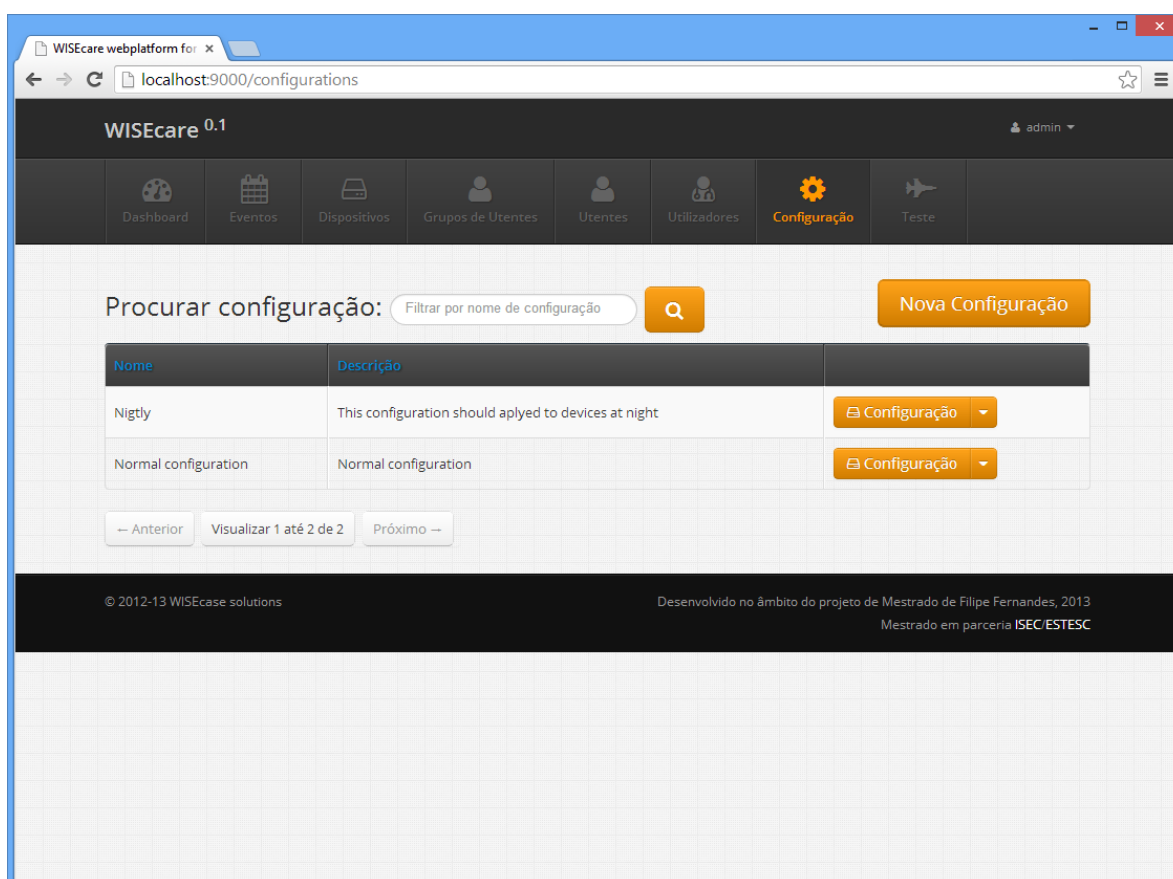


Figura 33 - Configuração

Nos ecrãs das figuras Figura 34 e Figura 35, pode configurar-se o nome e descrição do perfil e o valor de cada parâmetro.

As configurações não passam imediatamente para a *gateway*. São necessários alguns segundos para que as configurações sincronizem.

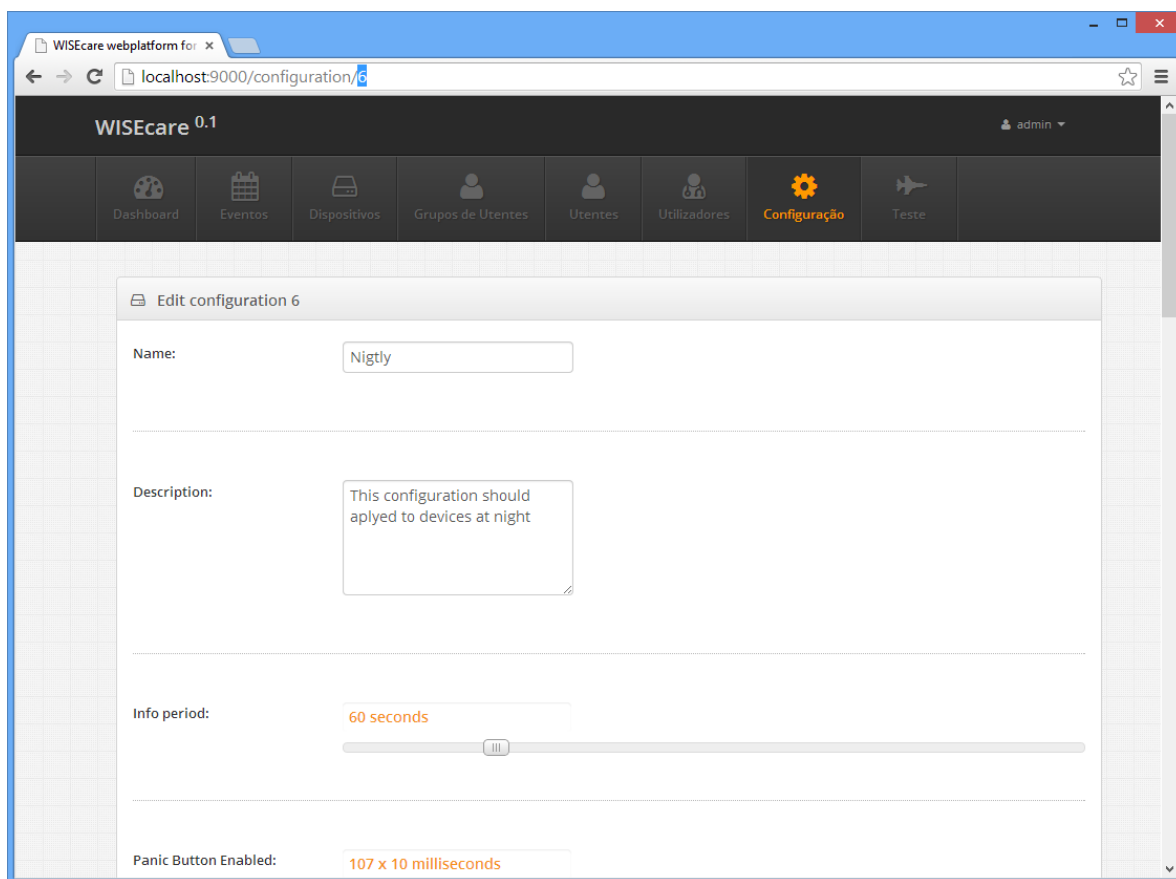


Figura 34 - Configuração – Edição 1/2

É, igualmente possível definir os valores dos diversos parâmetros de configuração do dispositivo.

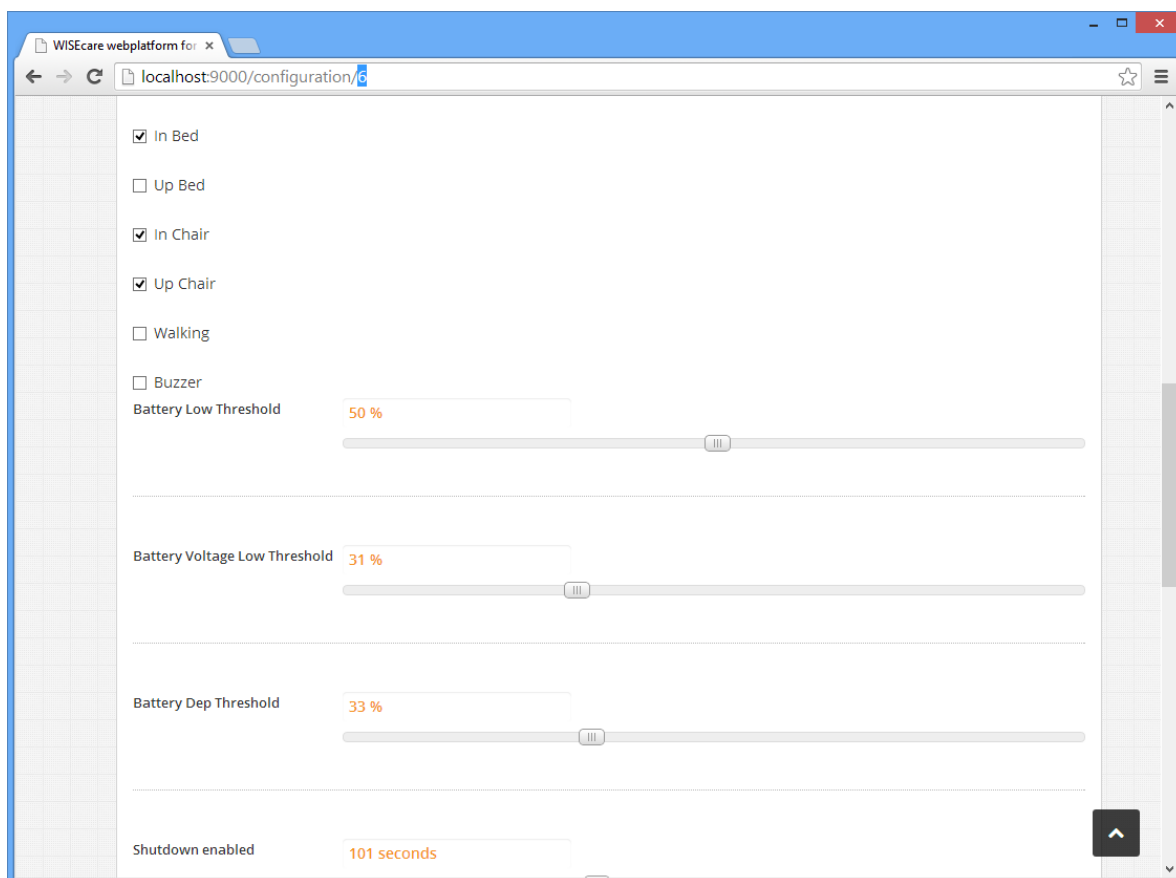


Figura 35 - Configuração - Edição 2/2

## 4.1.6.10 Alterar senha de utilizador

Existe um ecrã para o utilizador mudar a sua senha, Figura 36. Os utilizadores do tipo Administrador podem definir novas senhas para outros utilizadores.

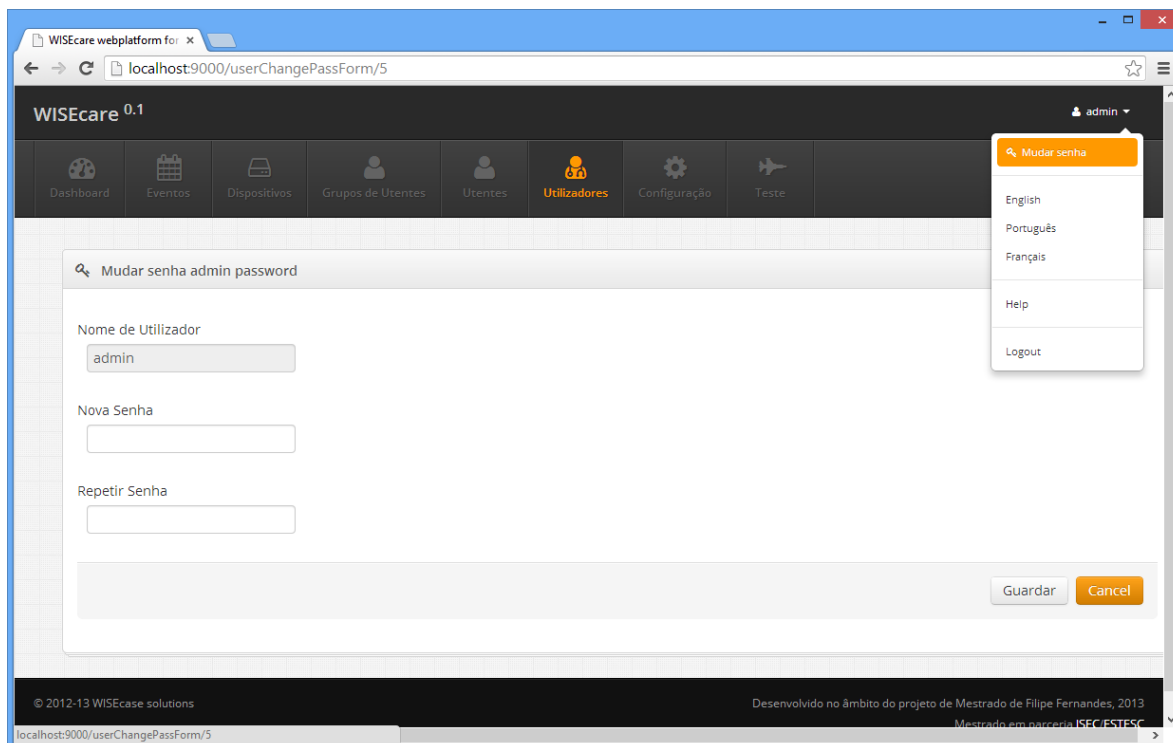


Figura 36 - Mudar senha

## 4.1.6.11 Suporte para multilíngue

Durante o desenvolvimento da plataforma foi tida em conta a possibilidade de suportar diversas línguas, Figura 37, pelo que todas as mensagens e todos os textos foram codificados de forma a ser possível traduzir a plataforma para diversas línguas.

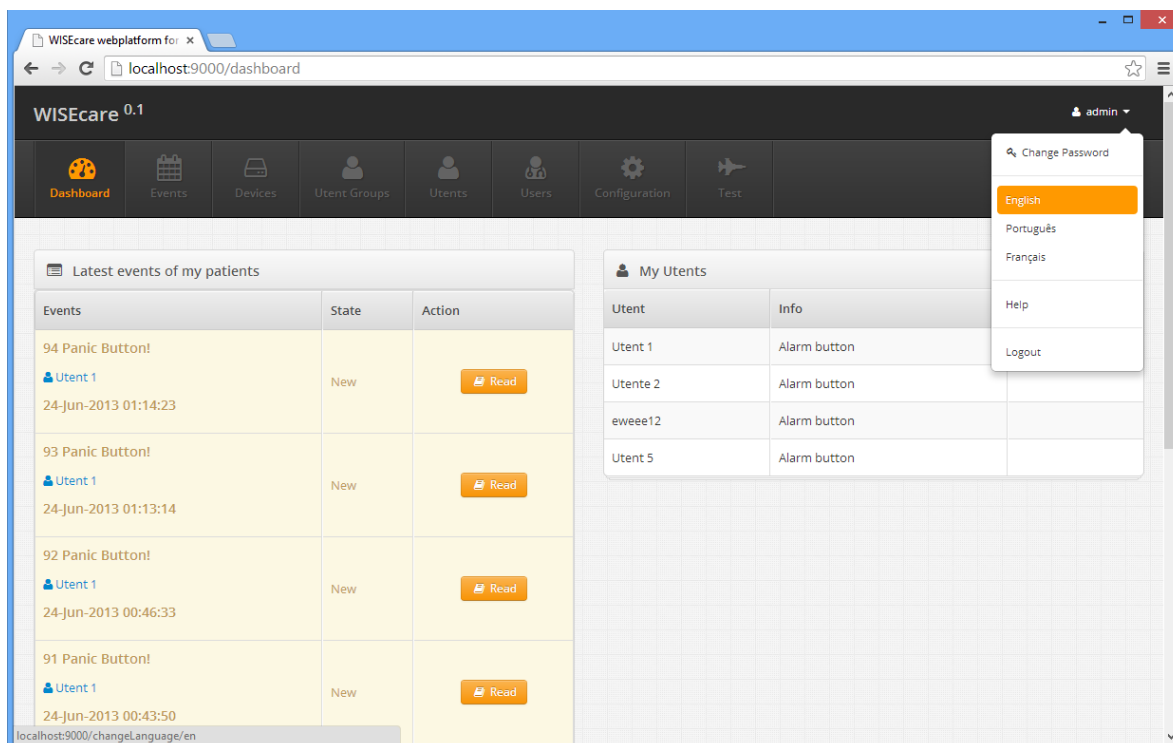


Figura 37 – Suporte para multilíngue (Plataforma WISEcare)

#### 4.1.6.12 Ajuda – Faqs

Na plataforma existe ajuda para o utilizador em forma de *faqs*, Figura 38. Na fase inicial foram explicadas apenas as funcionalidades, pretendendo-se no futuro incluir mais informação pertinente à medida que seja necessário. A ajuda encontra-se disponível em português e em inglês.

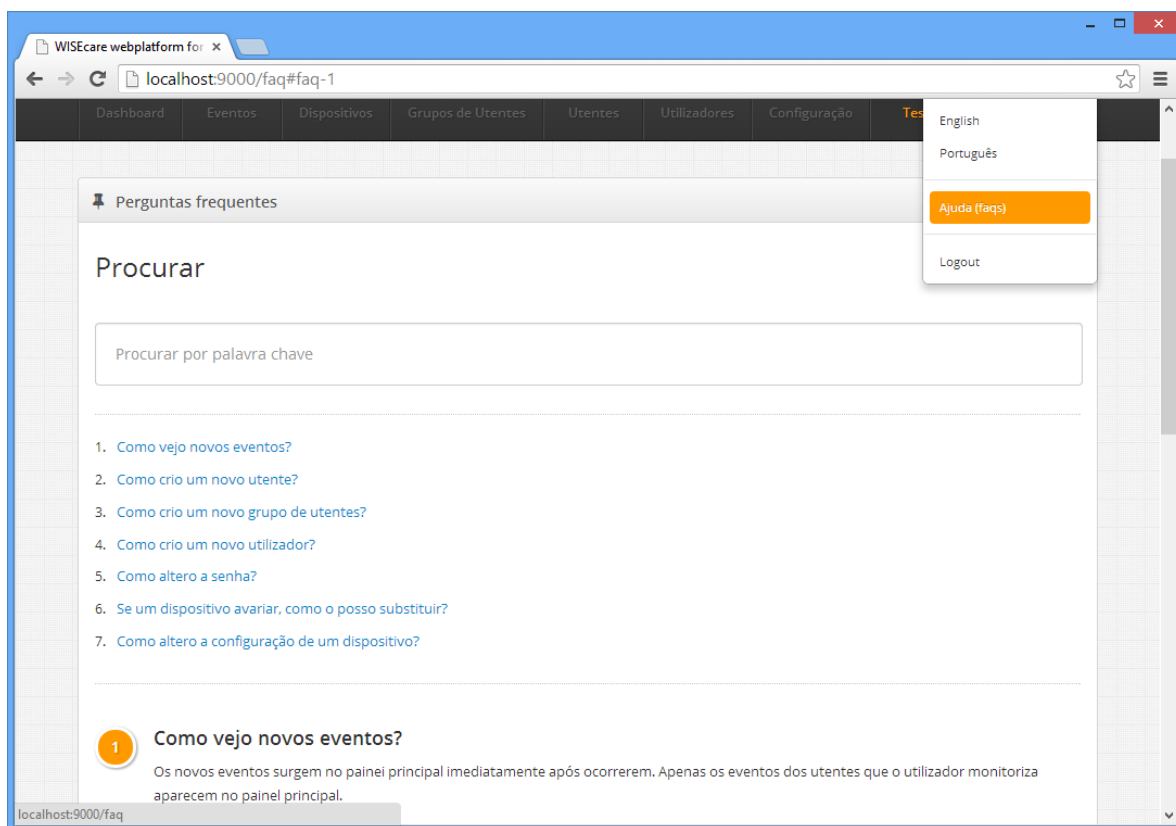


Figura 38 - Ajuda – Faqs (Plataforma WISEcare)



#### 4.1.6.13 Sistema de notificações na plataforma

Quando ocorre um alerta no sistema, surge uma mensagem com a descrição do alerta para os utilizadores responsáveis pelo utente.

Quando o utilizador visualizar o alerta, para além de o poder ler, tem a possibilidade de colocar uma descrição da intervenção realizada, ficando esta associada ao alerta para posterior consulta.

Caso o alerta tenha sido mostrado a vários utilizadores ligados ao sistema, quando o primeiro utilizador lê o alerta e coloca uma descrição, o alerta é eliminado dos restantes utilizadores. Esta funcionalidade evita que sejam tomadas ações repetidas sobre o mesmo evento.

O servidor envia mensagens para os clientes em formato Json com as seguintes estruturas: por exemplo, como ilustrado nas figuras Figura 39- Nova mensagem (Json) e Figura 40 - Mensagem lida (Json).

```
{
  "MsgType": "MessageNew",
  " HtmlClassTr ": "",
  "id": "1"
}
```

Figura 39- Nova mensagem (Json)

```
{
  "MsgType": "MessageReaded",
  " eventID ": "1",
}
```

Figura 40 - Mensagem lida (Json)

No momento em que é necessário obter novos dados do servidor, é feita uma chamada por *ajax* ao servidor para obter a nova informação sobre os eventos e sobre os utentes a mostrar no *dashboard*.

Existe uma codificação de tipos de alerta pré definidos com cores associadas, como ilustra a Tabela 15. Desta forma, Figura 41, torna-se mais simples para o monitor visualizar o nível de severidade da situação e tomar uma ação mais rápida sobre a ocorrência.

Tipo de evento	Nível de severidade	Hex	Cor associada
Utent Moving	0	#BAF3C3	
Out Of Range	1	#FFCBCF	
Panic Button	2	#F9FC9D	
	3	#DDE5FE	
	4	#D9B2FF	

Tabela 15 - Codificação dos níveis de severidade

Para a implementação desta funcionalidade, o HTML é manipulado por *JavaScript*, permitindo mostrar ou ocultar a mensagem de alerta.

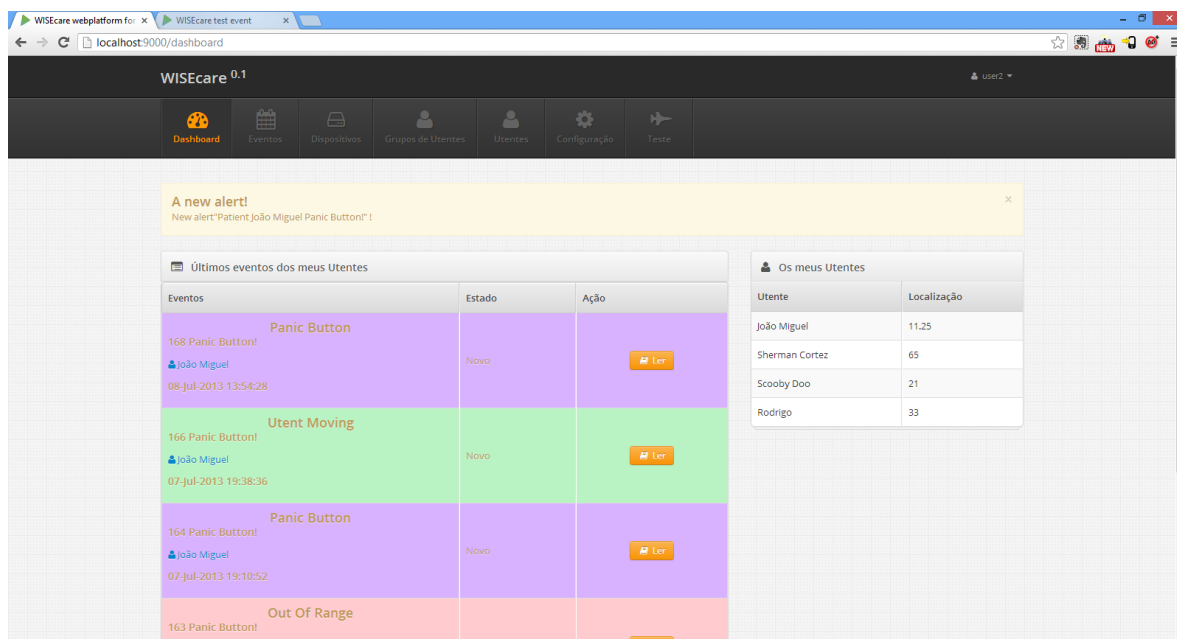


Figura 41 - Sistema de alertas

## 4.1.6.14 Colocação em produção

A vantagem de utilização da *play framework* é a possibilidade de colocar a aplicação na *cloud*, e usufruir de todas as vantagens de uma aplicação disponibilizada neste tipo de suporte.

Salientam-se entre muitas vantagens:

- Baixo custo inicial – normalmente custo zero durante a fase de testes;
- Sem necessidade de investimento em *hardware* ou *software*;
- Simplicidade de configuração;
- Alta disponibilidade e fiabilidade;
- Elasticidade de computação – Se eventualmente for necessário, é possível aumentar a capacidade do sistema sem necessidade de alterações ou novos desenvolvimentos.

Para efeitos de demonstração a aplicação foi colocada num servidor ec2, um serviço *cloud* da *amazon* e está disponível uma versão de demonstração em <http://wisecare.pt.vu>.

## 4.1.7 Implementação da Gateway WISEcare

A componente de *gateway* tem o objetivo de fazer de interface entre os dispositivos e a plataforma online. Um dos requisitos do *gateway* é o funcionamento de forma independente, isto é, sem necessidade de estar ligado à plataforma mas mesmo assim fornecendo as funcionalidades de alertas.

Além deste requisito, também é necessário criar um interface para gerir as configurações do *gateway*, um interface para registar e gerir dispositivos e suas configurações, bem como um painel para mostrar o estado geral do sistema, com o estado de cada dispositivo, os alertas gerados e gestão desses alertas.

A *gateway* tem a responsabilidade de esperar por ligações UDP num porto definido e quando recebe uma ligação tem de interpretar os dados recebidos, efetuando o respetivo tratamento segundo um protocolo definido pela WISEware, Lda. Os dados recebidos contêm informações sobre o estado do dispositivo, tais como o estado do botão de pânico, o estado de queda, o estado da “em cama”, “em cadeira”, os AP’s aos quais o dispositivo consegue captar sinal, etc.

Tem também a responsabilidade de verificar se o dispositivo está a enviar informação ou está fora de alcance (sem comunicação).

Optou-se por implementar a *gateway* igualmente na *play framework*, usufruindo assim de todas as funcionalidades anteriormente explicadas.

#### 4.1.7.1 Interface

##### 4.1.7.1.1 Painel principal

O painel principal da *Gateway WISEcare*, Figura 42, mostra a lista de dispositivos registados no sistema, estando associado a cada um informação sobre o estado da bateria, se está a ser usado e se está *online* atualmente.

Do lado direito mostra a lista dos últimos eventos ocorridos, havendo a possibilidade de esconder o evento (indicar que já foi lido).

O acesso aos parâmetros do dispositivo é feito através da opção de “Configuração” e é, ainda possível, desativar/ativar um dispositivo.

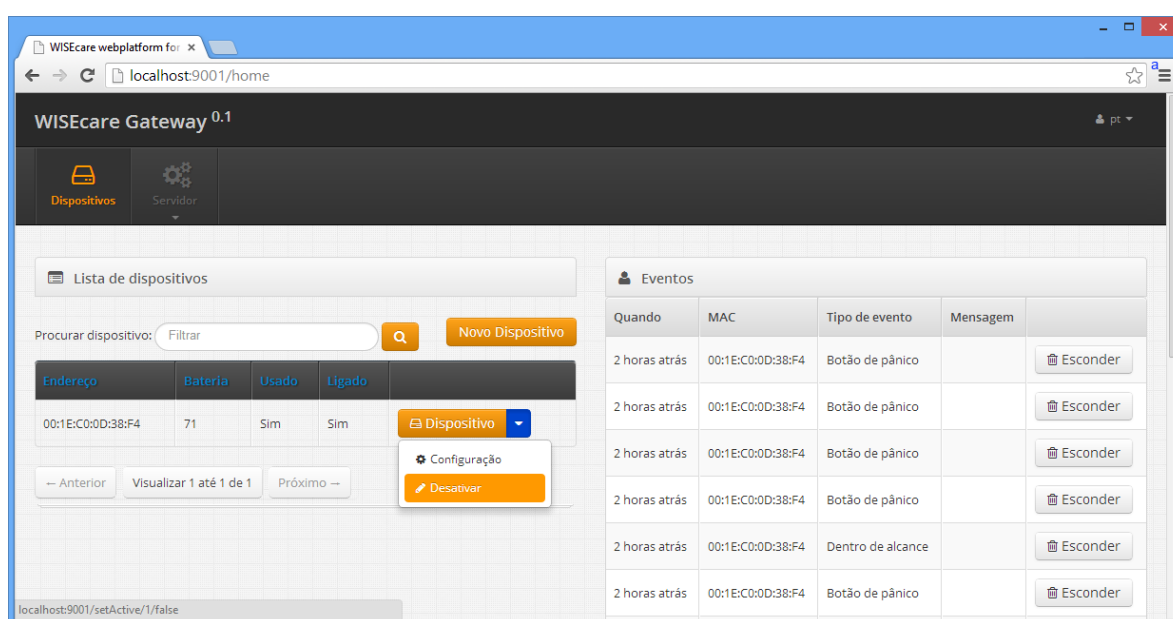


Figura 42 - Painel principal da Gateway WISEcare

#### 4.1.7.1.2 Configuração do Gateway WISEcare

No menu acede-se à configuração surgindo uma página com as definições atuais, Figura 43. Existe um botão para editar a configuração.

Este menu tem as opções para iniciar e parar o serviço. Só quando o serviço está em execução é que processa os eventos. Neste ecrã é possível visualizar o estado atual do serviço: Parado ou em execução.

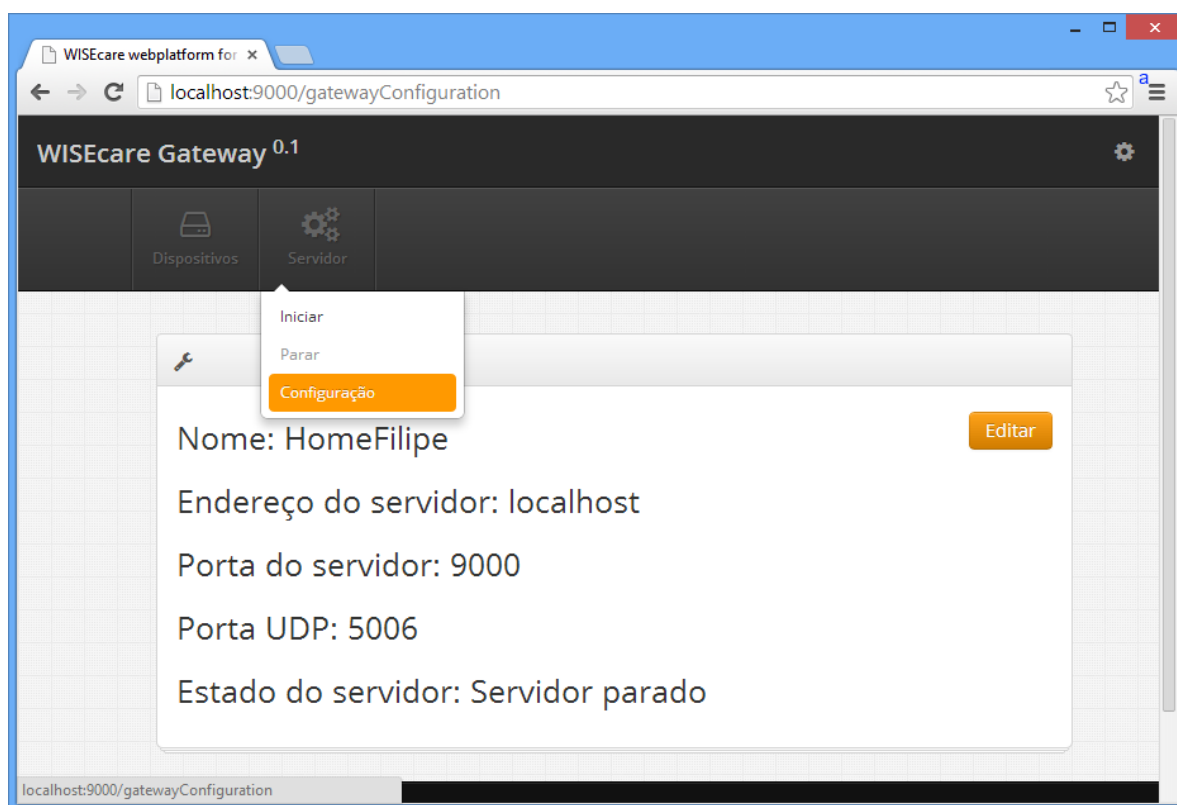


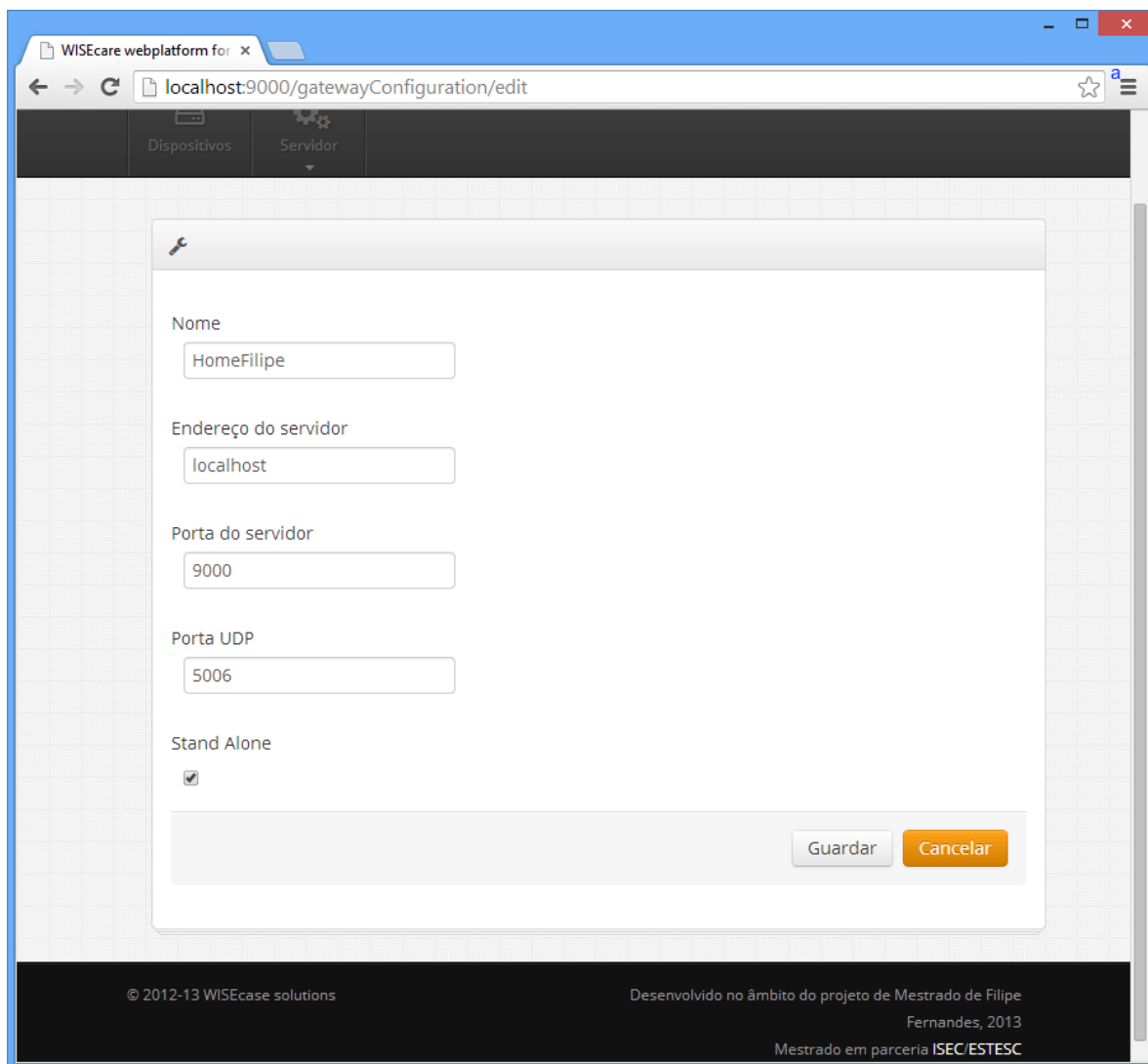
Figura 43 – Configuração do Gateway WISEcare

## 4.1.7.1.3 Editar configuração

Na página de edição de configuração, Figura 44, podem alterar-se os seguintes parâmetros do *gateway*:

- **Nome:** Nome para a Gateway. No caso de existirem várias *gateways* ligadas ao mesmo servidor, este nome é útil para saber a origem dos eventos;
- **Endereço do servidor:** nome qualificado ou endereço IP do servidor da *Plataforma WISEcare*;
- **Porta:** Porta onde o servidor se encontra a correr;
- **Porta UDP:** Porta de comunicação com os dispositivos. Protocolo UDP;
- **Stand alone:** Indica se o Gateway está a funcionar como *Stand alone*, isto é, não está ligado a um servidor. Se esta opção estiver ativa, a Gateway não obtém as configurações dos dispositivos do servidor e também não envia os eventos para o servidor.





**Figura 44 - Editar configuração**

#### 4.1.7.1.4 Parâmetros dos dispositivos

Os parâmetros dos dispositivos são semelhantes aos que foram descritos na secção de implementação da *Plataforma WISEcare*.

Neste interface, Figura 45, também é mostrado o tipo de configuração da Gateway (*stand alone*/ligada ao servidor) e o endereço MAC do dispositivo que está a ser configurado.

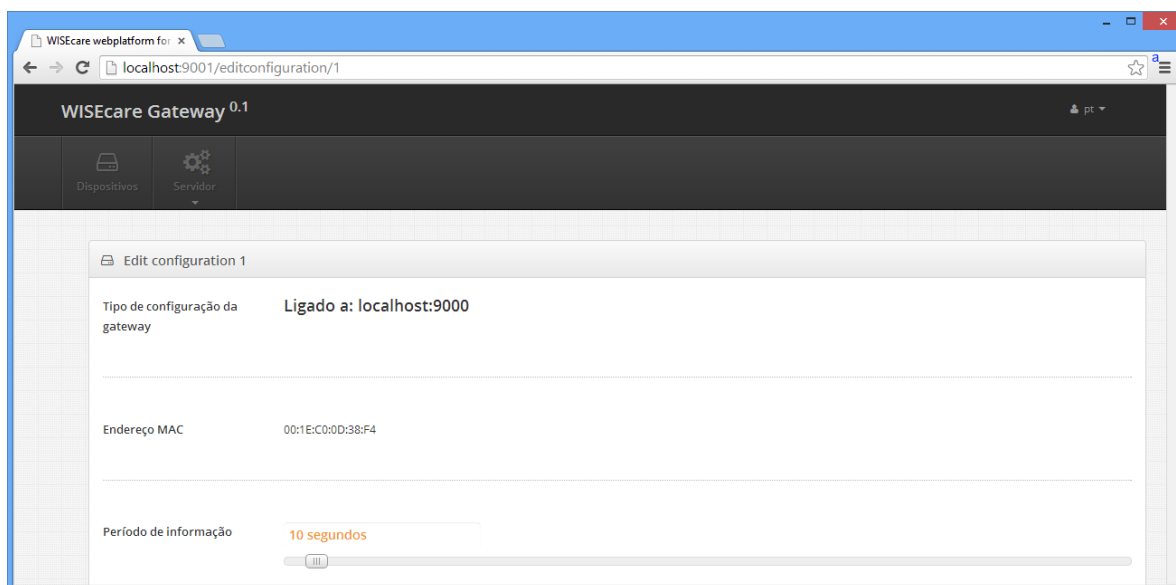


Figura 45 - Parâmetros dos dispositivos

## 4.1.7.1.5 Suporte para multilíngue

O interface da *gateway* encontra-se disponível em português e em inglês para o utilizador. Para alterar o idioma em uso, existe uma opção no canto superior direito com a lista dos idiomas disponíveis, Figura 46.

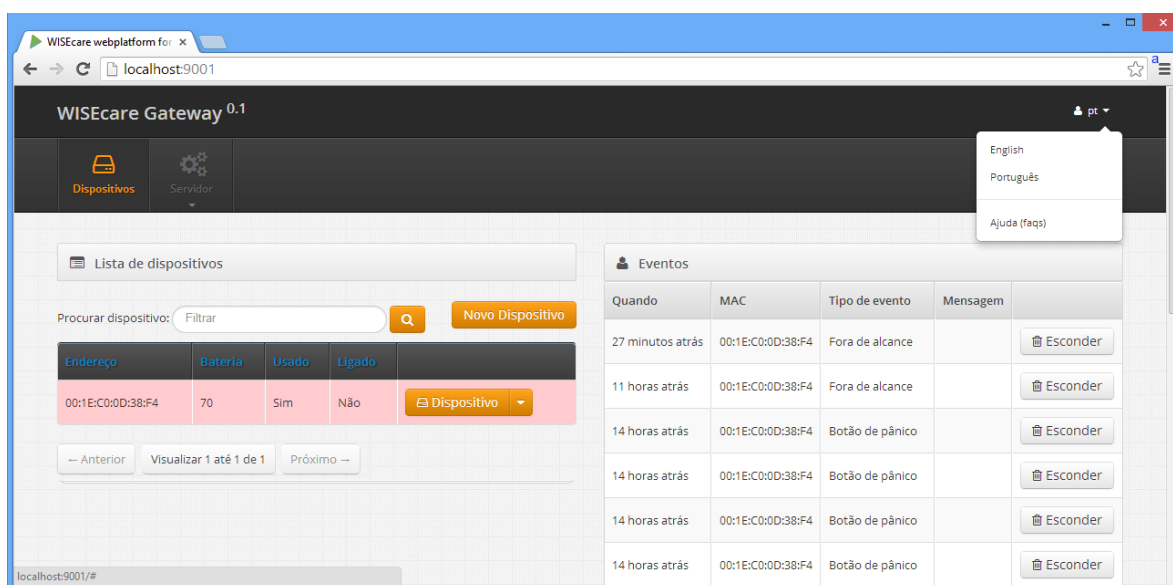


Figura 46 - Suporte para multilíngue (Gateway WISEcare)

## 4.1.7.1.6 Ajuda – Faqs

Existe, igualmente, uma página que contém informação sobre perguntas e respostas frequentes para ajudar o utilizador, Figura 47.

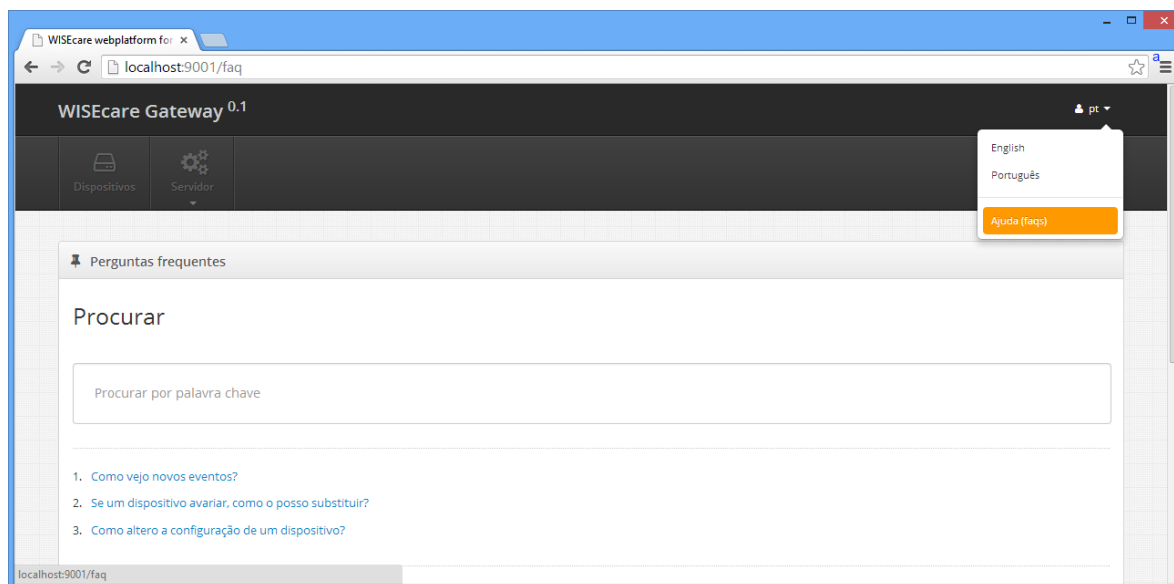


Figura 47 – Ajuda - Faqs (*Gateway WISEcare*)

#### 4.1.7.2 Sincronização de configurações

Quando o serviço de *gateway* inicia, é feita uma transferência “fresca” de todas as configurações dos dispositivos (no caso de não estar configurado como *stand alone*).

Ao longo do funcionamento, periodicamente é verificado se existem configurações novas a serem transferidas.

Para sincronizar as configurações é efetuado um pedido HTTP ao servidor (*post*) com uma mensagem em formato *Json*.

A mensagem enviada para o servidor tem a seguinte estrutura (exemplo na Figura 48):

```
{
  "MsgType": 1,
  "GatewayName": "HomeFilipe",
  "MACDevices": [
    {
      "MAC": "00:1E:C0:0D:38:F4",
      "configuration_version": 17
    }
  ]
}
```

Figura 48 - Mensagem Json pedido configurações

Podem ocorrer dois cenários:

A configuração da configuração no servidor é igual. O servidor devolve a seguinte mensagem, Figura 49:

```
{
  "MsgType": 2,
  "GatewayName": "HomeFilipe",
  "Devices": []
}
```

Figura 49 - Mensagem Json resposta configurações (sem configuração)

A versão da configuração no servidor é superior, e o servidor devolve a mensagem da Figura 50:

```
{
  "MsgType": 2,
  "GatewayName": "HomeFilipe",
  "Devices": [
    {
      "MAC": "00:1E:C0:0D:38:F4",
      "configuration_version": 18,
      "Configs": [
        {
          "amountINFO_PERIOD": "40"
        },
        ...
        {
          "validateCheckboxBUZZER": ""
        }
      ]
    }
  ]
}
```

Figura 50 - Mensagem Json envio configurações

Caso o servidor responda com uma versão superior, é guardada a configuração na tabela *params* e a respetiva versão na tabela *device*.

#### 4.1.7.3 Envio de eventos para a Plataforma WISEcare

As figuras Figura 51, Figura 52 e Figura 53 representam as mensagens que são enviadas para o servidor quando ocorre um evento.

```
{
  "description": "",
  "event_type": 1,
  "localization": "",
  "macAdress": "00:1E:C0:0D:38:F4",
  "batteryLevel": 100,
  "batteryCharging": true
}
```

Figura 51 - Mensagem Json envio evento para o servidor

O servidor devolve a seguinte mensagem Json:

```
{
  "msg": "Event added",
  "error": 0,
  "error_message": ""
}
```

Figura 52 - Mensagem resposta Json após inserir evento

Caso exista algum erro, este é devolvido nos campos “error” e “error\_message”.

```
{
  "msg": "Error adding event!",
  "error": 1,
  "error_message": "Device "00:1E:C0:0D:38:F4" not registered!"
}
```

Figura 53 - Mensagem Json com erro após inserir evento

#### 4.1.7.4 Sistema de notificações

Tal como na *Plataforma WISEcare*, a *gateway* também necessita de informar os utilizadores que ocorreram alertas no sistema. Também é usado um *websocket* para enviar informação para os clientes (*browser*) ligados ao sistema.

A informação enviada também vai em formato Json como mostra a Figura 54:

```
{
  "MsgType": "NewEvent",
  "EventType": 1,
  "adress": "00:1E:C0:0D:38:F4",
  "TrContentDeviceHtml": ""
}
```

Figura 54 - Mensagem Json com envio de um evento para um cliente web (browser)

Nota: Na mensagem enviada, segue logo o *html* formatado com o estado do *device* (*html* do elemento <tr>).



#### 4.1.7.5 Instalação

Este ponto mostra a instalação e configuração da Plataforma WISEcare e do Gateway WISEcare em ambiente Windows.

Pré requisitos:

- **Motor de base de dados MySQL** – É necessário instalar o motor de base dados MySQL com as configurações standard. Pode ser obtido em <http://www.mysql.com/downloads/>

Posteriormente é necessário criar uma base de dados com o nome “wisecare”.

Recomenda-se definir uma *password* e configurá-la no ficheiro “WISEcare\_play2.1.1\conf\application.conf” nas linhas:

db.default.user=

db.default.password=

No caso do motor de base de dados residir na mesma máquina, não é necessário configurar mais nada relativamente à base de dados. Caso se pretenda uma base de dados com um nome diferente ou colocar o motor em um servidor diferente, será necessário configurar o nome da base de dados e o nome do servidor no ficheiro de configurações *application.conf*.

- **Java:** É necessário instalar a última versão do jdk. A atual é a 7u25. Pode ser obtida em:  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

- **Play framework!** É necessário instalar a versão 2.1.1 da *framework*.

Pode ser obtida em <http://www.playframework.com/download>

Extrair o conteúdo do zip para C:\play-2.1.1

Colocar este caminho na variável de ambiente PATH, como ilustra a Figura 55.



## 4.1.7.5.1 Plataforma WISEcare

Para iniciar o servidor web:

Navegar até ao projeto pela linha de comandos e iniciar o serviço com o comando

- play start, Figura 57

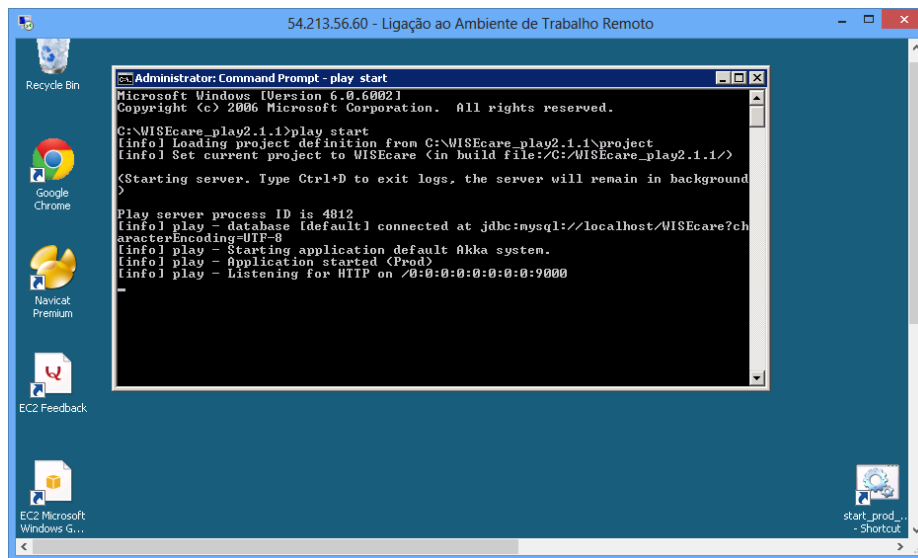
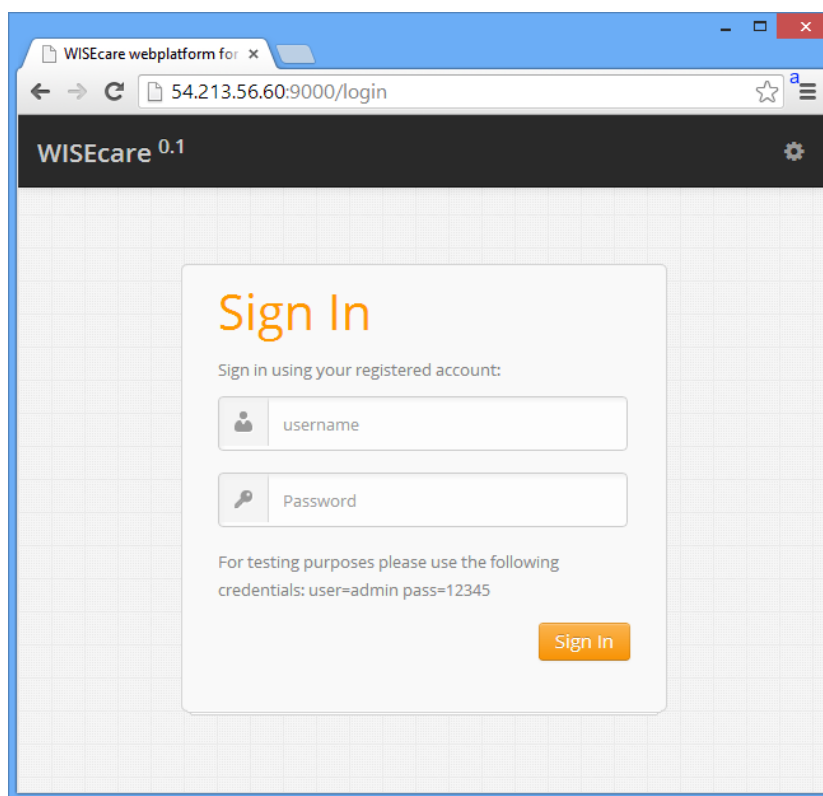


Figura 57 - Iniciar o serviço da *Plataforma WISEcare*

Após iniciar o serviço já é possível testar a plataforma, Figura 58, abrindo um *browser* e colocando o IP e porta do serviço. Normalmente a 9000.



**Figura 58 - Testar a plataforma**

## 4.1.7.5.2 Gateway WISEcare

O procedimento para a *gateway* é semelhante.

Executar o comando

- play start, Figura 59

```

Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Todos os direitos reservados.

C:\mestrado\WISEcare\trunk>cd WISEcareGateway_play2.1.1

C:\mestrado\WISEcare\trunk\WISEcareGateway_play2.1.1>play start
[info] Loading project definition from C:\mestrado\WISEcare\trunk\WISEcareGateway_play2.1.1\project
[info] Set current project to WISEcareGateway (in build file:/C:/mestrado/WISEcare/trunk/WISEcareGateway_play2.1.1/)

<Starting server. Type Ctrl+D to exit logs, the server will remain in background>

Play server process ID is 9052
[info] play - database [default] connected at jdbc:mysql://localhost/WISEcareGateway?characterEncoding=UTF-8
[debug] application - On start
[info] play - Application started (Prod)
[info] play - Listening for HTTP on /0:0:0:0:0:0:0:0:9000
  
```

Figura 59 - Iniciar Gateway WISEcare

O serviço inicia e pode ser testado, Figura 60:

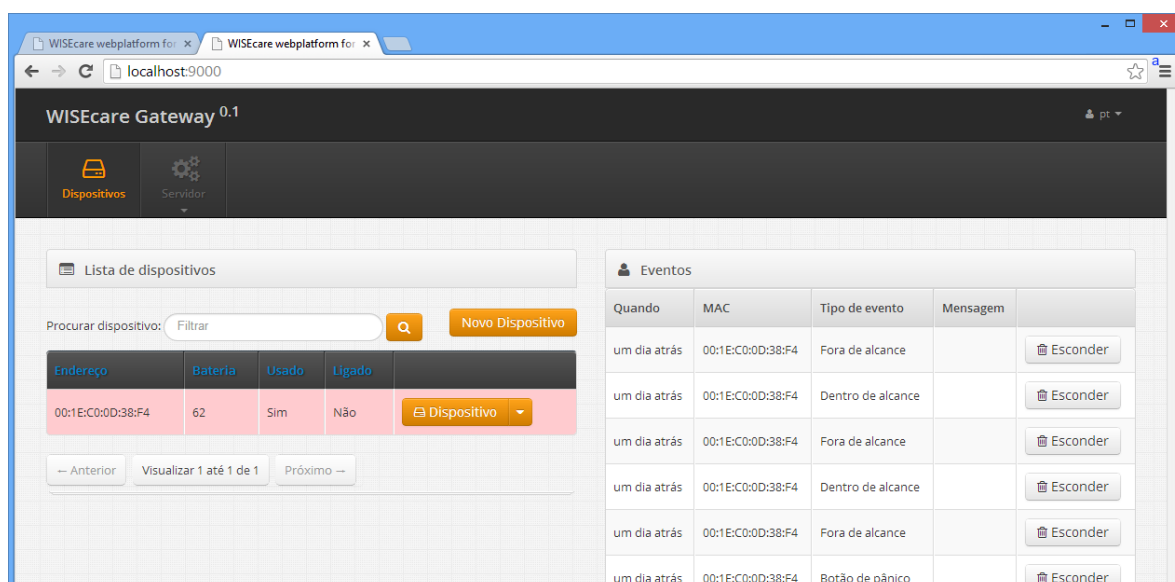


Figura 60 - Teste Gateway WISEcare

## Capítulo 5. Conclusão e desenvolvimentos futuros

No presente capítulo são expostos os resultados reais de todo o trabalho realizado. É apresentada uma síntese das conclusões e referidas as pistas de desenvolvimento mais importantes.

Em termos práticos a contribuição principal deste trabalho consiste na implementação de um sistema de informação que permita a monitorização de forma permanente e fiável de pessoas que necessitem de vigilância constante ou parcial.

A solução obtida possibilita a monitorização de um conjunto de utentes e permite uma intervenção rápida nas situações de emergência e/ou nos casos de pedido de auxílio no dia-a-dia de uma instituição de saúde.

Devo referir que grande parte da tecnologia usada, especialmente a que envolve a componente web, era uma área que eu não dominava, razão pela qual este projeto se tornou um desafio maior e mais interessante.

Todo o trabalho de análise realizado, com a identificação de cada necessidade encontrada, assim como, com o esquema do funcionamento pretendido, a identificação dos atores e dos dados que são tratados no sistema é descrito no Capítulo 3. Neste capítulo também se apresentam a estrutura de dados usada com descrições detalhadas sobre cada campo. São mostrados também os estudos iniciais de interface, no formato de *mockups*.

O Capítulo 4 refere-se à implementação do projeto realizado, falando-se no modo como este foi implementado, e quais as ferramentas e tecnologias usadas para se conseguir cada objetivo. São apresentadas todas as funcionalidades implementadas com *screenshots* e com uma explicação detalhada.

Os objetivos iniciais não eram rígidos e houve alguma liberdade para explorar e experimentar novas soluções. O presente trabalho possui as funcionalidades essenciais para o funcionamento da plataforma, cumprindo o objetivo inicial que era o de monitorizar pessoas com dificuldades motoras e emitir alertas em caso de necessidade.

Como se trata de um portal com registo de utilizadores e utentes, é fácil imaginar o seu uso para fazer toda a gestão necessária numa instituição que acolhe este tipo de pessoas. Existe portanto um potencial interesse em evoluir esta plataforma, acrescentando outras funcionalidades para facilitar os profissionais que atuam nesta área. Assim, considero que seria de grande utilidade o registo de mais informação relativa ao utente, como por exemplo: o registo de medicação receitada e administrada, registo de alimentação, com informação de dietas, registo das próximas consultas. Associado a estas funcionalidades, existe um vasto leque de informação que poderia ser extraída posteriormente, no sentido de melhorar a qualidade de vida dos utentes e de otimizar as tarefas dos profissionais de saúde.

O sistema nesta fase tem apenas funcionalidades de monitorização e alertas de utentes. Com a arquitetura montada será muito fácil escalar o sistema e implementar a recolha de informação biomédica relativa aos utentes monitorizados. Com essa informação recolhida, e dependendo do tipo de informação, o leque de possibilidades alarga-se. Surgem muitas possibilidades no campo de extração de informação, produção de relatórios. O sistema pode reagir automaticamente quando recebe um conjunto de valores fora do intervalo aceitável.

## Bibliografia

Abbate, S., Avvenuti, M., Corsini, P., Light, J., & Vecchio, A. (2010). Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network. INTECH.

Darkins, A., & Sanders, J. H. (6 de Abril de 2009). Remote Monitoring Expanding a Successful System. Nursing Management, p. Vol. 28i N. 5.

Gonçalves, P. R. (30 de Dezembro de 2008). Monitorização Remota de Pacientes em Ambulatório. Porto: UNIVERSIDADE FERNANDO PESSOA.

Rougier, C., St-Arnaud, A., Rousseau, J., & Jacqueline, J. (2011). Video Surveillance for Fall Detection, Video Surveillance. InTech.

Fontes online:

Ebean Wikipedia. (26 de Setembro de 2012). Obtido em 2013 de Maio de 24, de Wikipédia:

<http://en.wikipedia.org/wiki/Ebean>

Desenvolvimento ágil de software. (20 de Agosto de 2013). Obtido em 20 de Agosto de 2013, de Desenvolvimento ágil de software:

[http://pt.wikipedia.org/wiki/Desenvolvimento\\_%C3%A1gil\\_de\\_software](http://pt.wikipedia.org/wiki/Desenvolvimento_%C3%A1gil_de_software)

Javascript. (27 de Junho de 2013). Obtido em 27 de Junho de 2013, de

<http://pt.wikipedia.org/wiki/JavaScript>

MAC Adress. (24 de Junho de 2013). Obtido de

[http://pt.wikipedia.org/wiki/Endere%C3%A7o\\_MAC](http://pt.wikipedia.org/wiki/Endere%C3%A7o_MAC)

MYSQL. (1 de Maio de 2013). Obtido em 2013 de Maio de 22, de

<http://pt.wikipedia.org/wiki/MySQL>: <http://pt.wikipedia.org/wiki/MySQL>

Oracle. (27 de Junho de 2013). Oracle - Java. Obtido em 27 de Junho de 2013, de O que é a tecnologia Java e por que é necessária?:

[http://www.java.com/pt\\_BR/download/faq/whatis\\_java.xml](http://www.java.com/pt_BR/download/faq/whatis_java.xml)

ORM - Wikipedia. (9 de Maio de 2013). Obtido em 24 de Maio de 2013, de Wikipédia:

[http://en.wikipedia.org/wiki/Object-relational\\_mapping](http://en.wikipedia.org/wiki/Object-relational_mapping)

Padrão MVC - Wikipédia. (23 de Maio de 2013). Obtido em 2013 de Maio de 2013, de

Wikipédia: <http://pt.wikipedia.org/wiki/MVC>

Play framework! (24 de Maio de 2013). Obtido em 24 de Maio de 2013, de Play

Framework: <http://www.playframework.com/documentation/2.1.1/Home>



Templates Scala. (26 de Maio de 2013). Obtido em 26 de Maio de 2013, de Scala templates:

<http://www.playframework.com/documentation/2.0/ScalaTemplates>

Wikipedia - Scala. (7 de Abril de 2013). Obtido em 22 de Maio de 2013, de Wikipedia:

[https://pt.wikipedia.org/wiki/Scala\\_\(linguagem\\_de\\_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Scala_(linguagem_de_programa%C3%A7%C3%A3o))

WebSockets -Wikipedia . (s.d.). Obtido em 27 de Junho de 2013, de

<http://pt.wikipedia.org/wiki/WebSockets>

Wikipedia, M. (20 de Agosto de 2013). Obtido em 20 de Agosto de 2013, de Wikipedia:

<http://en.wikipedia.org/wiki/Mockup>

Wikipedia, P. m. (20 de Agosto de 2013). Wikipedia, Push message. Obtido em 20 de

Agosto de 2013, de [http://en.wikipedia.org/wiki/Push\\_message](http://en.wikipedia.org/wiki/Push_message)

Wikipedia, S. (20 de Agosto de 2013). SMS Wikipedia. Obtido em 27 de Agosto de 2013, de

SMS Wikipedia: [http://en.wikipedia.org/wiki/Short\\_Message\\_Service](http://en.wikipedia.org/wiki/Short_Message_Service)